What is a Proof?

Jean Gallier and
Kurt W.A.J.H.Y. Reillag
CIS, Upenn and
Hospices de Beaune

Reillag’s office

Another office

After a bad proof!

Finally, Reillag (young)

Quick History

Quick History

- Formalizing the rules of logic goes back to the Greek.

Quick History

- Formalizing the rules of logic goes back to the Greek.
- Axioms and Syllogisms (Aristotle, 384 BC-322 BC)
  - All humans are mortal
  - Socrates is a human
  - Socrates is mortal.

Quick History

- Formalizing the rules of logic goes back to the Greek.
- Axioms and Syllogisms (Aristotle, 384 BC-322 BC)
  - All humans are mortal
  - Socrates is a human
  - Socrates is mortal.
- Modus Ponens: If (P implies Q) holds and P holds, then Q holds.

Types of Proofs

Types of Proofs

- Proof by intimidation

Types of Proofs

- Proof by intimidation
- Proof by seduction

Types of Proofs

- Proof by intimidation
- Proof by seduction
- Proof by interruption

Types of Proofs

- Proof by intimidation
- Proof by seduction
- Proof by interruption
- Proof by misconception

Types of Proofs

- Proof by intimidation
- Proof by seduction
- Proof by interruption
- Proof by misconception
- Proof by obfuscation

Types of Proofs

- Proof by intimidation
- Proof by seduction
- Proof by interruption
- Proof by misconception
- Proof by obfuscation
- Proof by confusion

Types of Proofs

- Proof by intimidation
- Proof by seduction
- Proof by interruption
- Proof by misconception
- Proof by obfuscation
- Proof by confusion
- Proof by exhaustion

More Types of Proofs

More Types of Proofs

- Proof by passion

More Types of Proofs

- Proof by passion
- Proof by example

More Types of Proofs

- Proof by passion
- Proof by example
- Proof by vigorous handwaving

More Types of Proofs

- Proof by passion
- Proof by example
- Proof by vigorous handwaving
- Proof by cumbersome notation

More Types of Proofs

- Proof by passion
- Proof by example
- Proof by vigorous handwaving
- Proof by cumbersome notation
- Proof by omission

More Types of Proofs

- Proof by passion
- Proof by example
- Proof by vigorous handwaving
- Proof by cumbersome notation
- Proof by omission
- Proof by funding

More Types of Proofs

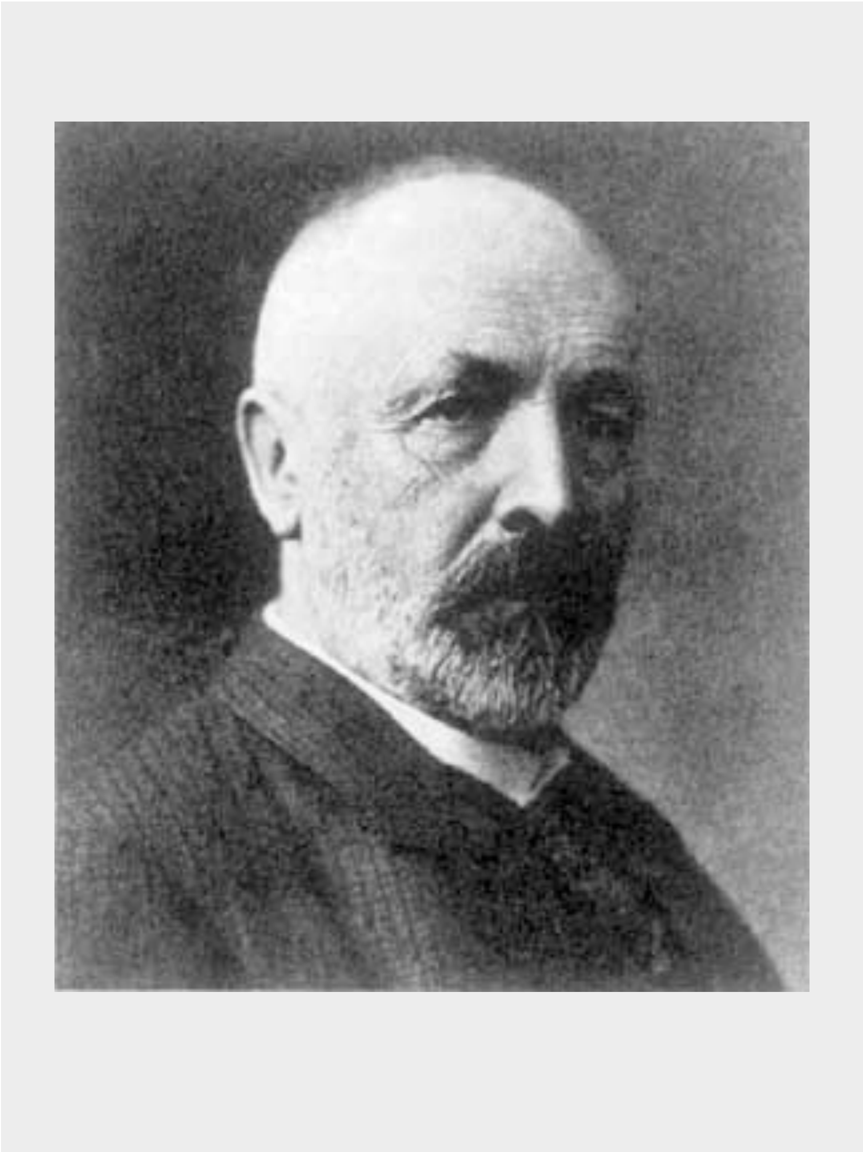
- Proof by passion
- Proof by example
- Proof by vigorous handwaving
- Proof by cumbersome notation
- Proof by omission
- Proof by funding
- Proof by personal communication

More Types of Proofs

- Proof by passion
- Proof by example
- Proof by vigorous handwaving
- Proof by cumbersome notation
- Proof by omission
- Proof by funding
- Proof by personal communication
- Proof by metaproof, etc.

Proof by intimidation!

Quick History

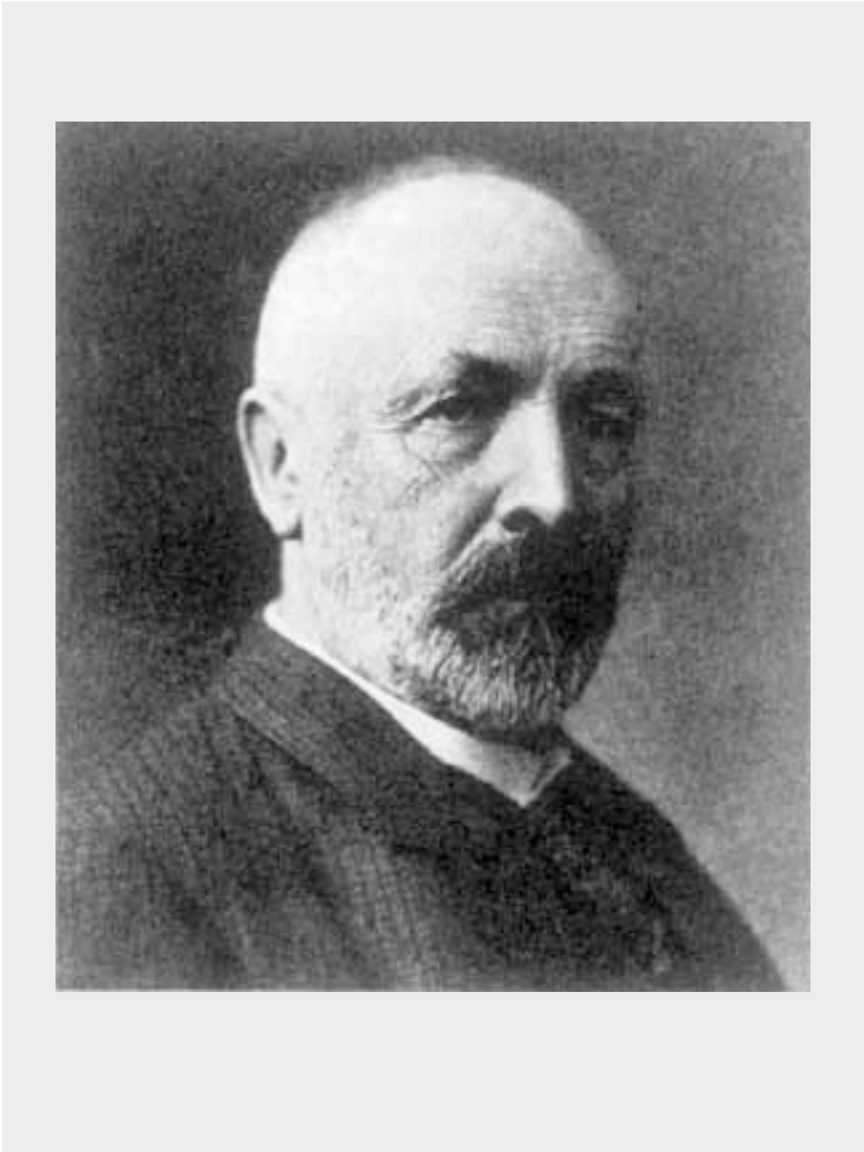


Quick History

- Cantor (1845-1918) and the birth of set theory

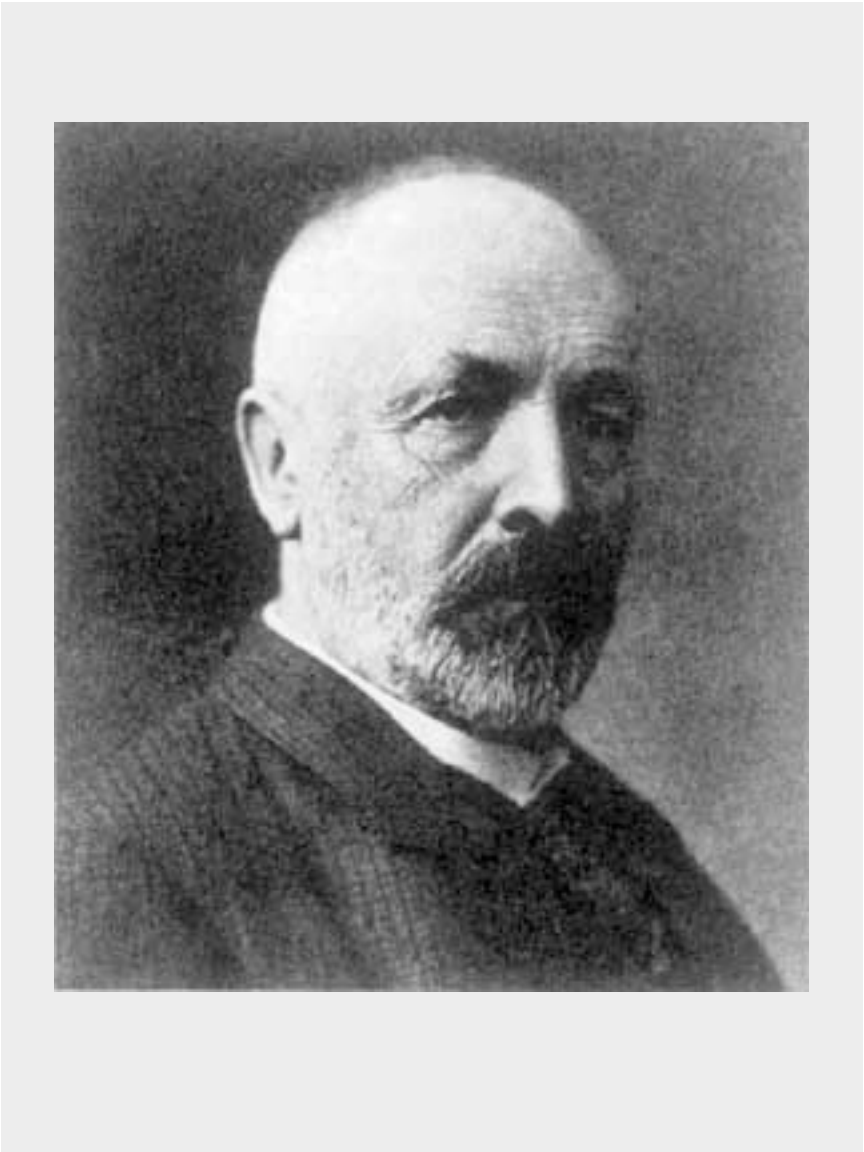
Quick History

- Cantor (1845-1918) and the birth of set theory
- Paradoxes and the "crisis of foundations".



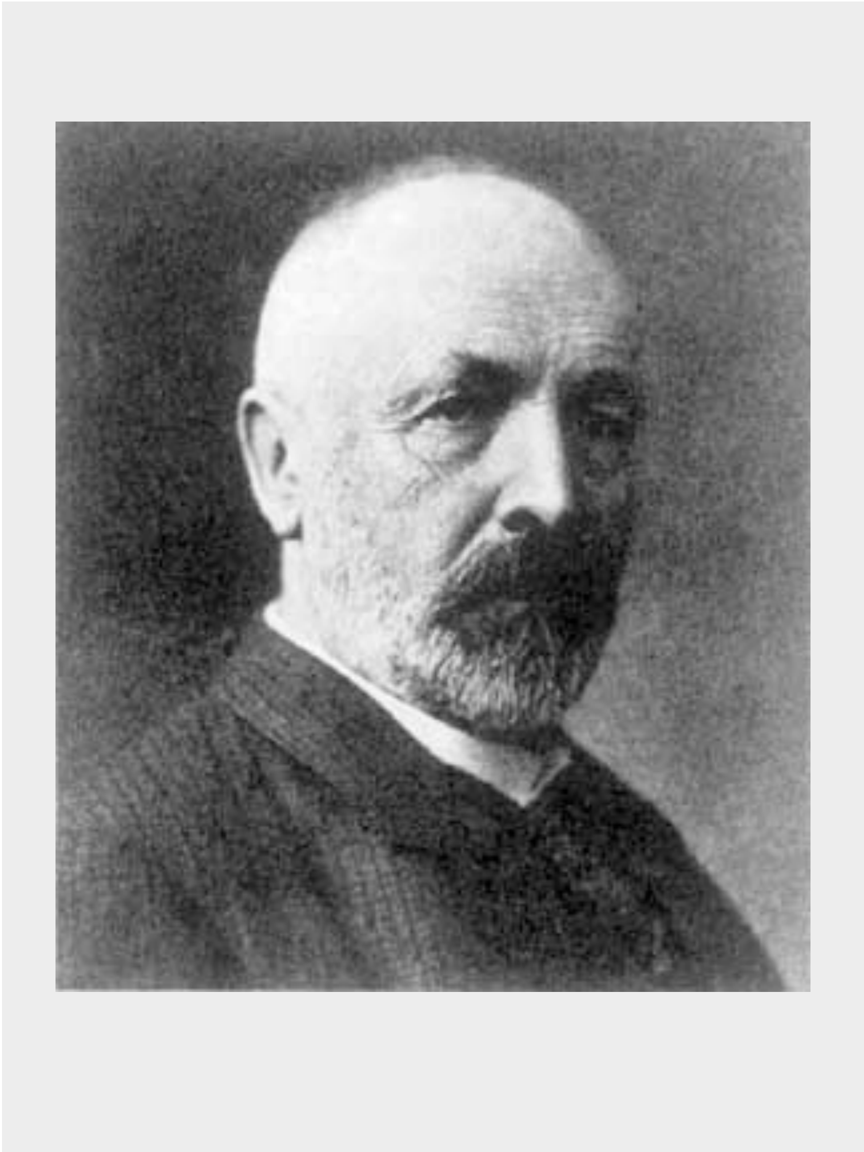
Quick History

- Cantor (1845-1918) and the birth of set theory
- Paradoxes and the "crisis of foundations".
- Sets that are too big or defined by self-reference



Quick History

- Cantor (1845-1918) and the birth of set theory
- Paradoxes and the "crisis of foundations".
- Sets that are too big or defined by self-reference
- Russell's paradox (1902)



Quick History

- Cantor (1845-1918) and the birth of set theory
- Paradoxes and the "crisis of foundations".
- Sets that are too big or defined by self-reference
- Russell's paradox (1902)
- There is no set of all sets

Truth and Proofs

Truth and Proofs

- Ideally, we would like to know what is truth

Truth and Proofs

- Ideally, we would like to know what is truth
- From the point of view of logic, truth has to do with semantics, i.e., the meaning of statements

Truth and Proofs

- Ideally, we would like to know what is truth
- From the point of view of logic, truth has to do with semantics, i.e., the meaning of statements
- Peter Andrew's motto: ``Truth is elusive"

Truth and Proofs

- Ideally, we would like to know what is truth
- From the point of view of logic, truth has to do with semantics, i.e., the meaning of statements
- Peter Andrew's motto: ``Truth is elusive"
- ``To truth through proof"

Truth and Proofs

- Ideally, we would like to know what is truth
- From the point of view of logic, truth has to do with semantics, i.e., the meaning of statements
- Peter Andrew's motto: ``Truth is elusive"
- ``To truth through proof"
- Provable implies true. Easier to study proofs

Hilbert

David Hilbert (1862-1943)

Hilbert Systems

Hilbert Systems

- Hilbert systems have many axioms and few inference rules

Hilbert Systems

- Hilbert systems have many axioms and few inference rules
- The axioms are very unnatural!

Hilbert Systems

- Hilbert systems have many axioms and few inference rules
- The axioms are very unnatural!
- That's because they are chosen to yield the deduction theorem

Hilbert Systems

- Hilbert systems have many axioms and few inference rules
- The axioms are very unnatural!
- That's because they are chosen to yield the deduction theorem
- Unfriendly system for humans.

Hilbert Systems

- Hilbert systems have many axioms and few inference rules
- The axioms are very unnatural!
- That's because they are chosen to yield the deduction theorem
- Unfriendly system for humans.
- Proofs in Hilbert systems are very far from proofs that a human would write

Gentzen's Systems

Gentzen's Systems

- Gerhard Gentzen (1909-1945)

Gentzen's Systems

- Gerhard Gentzen (1909-1945)
- Introduced natural deduction systems and sequent calculi

Gentzen's Systems

- Gerhard Gentzen (1909-1945)
- Introduced natural deduction systems and sequent calculi
- Trivial axioms, ``natural rules"

Gentzen's Systems

- Gerhard Gentzen (1909-1945)
- Introduced natural deduction systems and sequent calculi
- Trivial axioms, ``natural rules"
- The rules formalize informal rules of reasoning

Gentzen's Systems

- Gerhard Gentzen (1909-1945)
- Introduced natural deduction systems and sequent calculi
- Trivial axioms, ``natural rules"
- The rules formalize informal rules of reasoning
- Symmetry of the rules

Gentzen's Systems

- Gerhard Gentzen (1909-1945)
- Introduced natural deduction systems and sequent calculi
- Trivial axioms, ``natural rules"
- The rules formalize informal rules of reasoning
- Symmetry of the rules
- Introduction/Elimination

Proofs and Deductions

Proofs and Deductions

- A proof of a proposition, P, does not depend on any assumptions (premises).

Proofs and Deductions

- A proof of a proposition, P, does not depend on any assumptions (premises).
- When we construct a proof, we usually introduce extra premises which are later closed (dismissed, discharged).

Proofs and Deductions

- A proof of a proposition, P, does not depend on any assumptions (premises).
- When we construct a proof, we usually introduce extra premises which are later closed (dismissed, discharged).
- Such an ``unfinished” proof is a deduction.

Proofs and Deductions

- A proof of a proposition, P, does not depend on any assumptions (premises).
- When we construct a proof, we usually introduce extra premises which are later closed (dismissed, discharged).
- Such an ``unfinished” proof is a deduction.
- We need a mechanism to keep track of closed (discharged) premises (the others are open).

Natural Deduction Rules

- A proof is a tree labeled with propositions
- To prove an implication, $P \Rightarrow Q$, from a list of premises, $\Gamma = (P_1, \ldots, P_n)$, do this:
- Add P to the list Γ and prove Q from Γ and P.
- When this deduction is finished, we obtain a proof of $P \Rightarrow Q$ which does not depend on P, so the premise P needs to be discharged (closed).

Natural Deduction Rules

The axioms and inference rules for *implicational logic* are:

Axioms:

$$\frac{\Gamma, P}{P}$$

The $\Rightarrow$-*elimination rule*:

$$\frac{\dfrac{\Gamma}{P \Rightarrow Q} \qquad \dfrac{\Delta}{P}}{Q}$$

Natural Deduction Rules

The $\Rightarrow$-*introduction rule*:

$$\cfrac{\cfrac{\Gamma, P^x}{Q}}{P \Rightarrow Q}\ x$$

In the introduction rule, the tag x indicates which rule caused the premise, P, to be discharged.

Natural Deduction Rules

The $\Rightarrow$-*introduction rule*:

$$\cfrac{\cfrac{\Gamma, P^x}{Q}}{P \Rightarrow Q}\ x$$

In the introduction rule, the tag x indicates which rule caused the premise, P, to be discharged.

Every tag is associated with a unique rule but several premises can be labeled with the same tag and all discharged in a single step.

Examples of Proofs

(a)

$$\cfrac{\cfrac{P^x}{P}}{P \Rightarrow P}\; x$$

So, $P \Rightarrow P$ is provable; this is the least we should expect from our proof system!

(b)

$$\cfrac{\cfrac{\cfrac{\cfrac{(Q \Rightarrow R)^y \qquad \cfrac{(P \Rightarrow Q)^z \qquad P^x}{Q}}{R}}{P \Rightarrow R}\; x}{(Q \Rightarrow R) \Rightarrow (P \Rightarrow R)}\; y}{(P \Rightarrow Q) \Rightarrow ((Q \Rightarrow R) \Rightarrow (P \Rightarrow R))}\; z$$

Examples of proofs

(c) In the next example, the two occurrences of A labeled x are discharged simultaneously.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(A \Rightarrow (B \Rightarrow C))^z \qquad A^x}{B \Rightarrow C} \qquad \dfrac{(A \Rightarrow B)^y \qquad A^x}{B}}{C}}{A \Rightarrow C}\, x}{(A \Rightarrow B) \Rightarrow (A \Rightarrow C)}\, y}{\big(A \Rightarrow (B \Rightarrow C)\big) \Rightarrow \big((A \Rightarrow B) \Rightarrow (A \Rightarrow C)\big)}\, z
$$

More Examples of Proofs

(d) In contrast to Example (c), in the proof tree below the two occurrences of A are discharged separately. To this effect, they are labeled differently.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(A \Rightarrow (B \Rightarrow C))^z \qquad A^x}{B \Rightarrow C} \qquad \dfrac{(A \Rightarrow B)^y \qquad A^t}{B}}{C}}{A \Rightarrow C}\, x}{(A \Rightarrow B) \Rightarrow (A \Rightarrow C)}\, y}{\big(A \Rightarrow (B \Rightarrow C)\big) \Rightarrow \big((A \Rightarrow B) \Rightarrow (A \Rightarrow C)\big)}\, z}{A \Rightarrow \Big(\big(A \Rightarrow (B \Rightarrow C)\big) \Rightarrow \big((A \Rightarrow B) \Rightarrow (A \Rightarrow C)\big)\Big)}\, t
$$

Wow, I landed it! (the proof)

Natural Deduction in Sequent-Style

- A different way of keeping track of open premises (undischarged) in a deduction
- The nodes of our trees are now **sequents** of the form $\Gamma \to P$, with

  $$\Gamma = x_1 : P_1, \ldots, x_m : P_m$$

- The variables are pairwise distinct but the premises may be repeated
- We can view the premise P_i as the **type** of the variable x_i!

Natural Deduction in Sequent-Style

The *axioms and rules for implication in Gentzen-sequent style*:

$$\Gamma, x\colon P \rightarrow P$$

$$\frac{\Gamma, x\colon P \rightarrow Q}{\Gamma \rightarrow P \Rightarrow Q} \quad (\Rightarrow\text{-}intro)$$

$$\frac{\Gamma \rightarrow P \Rightarrow Q \quad \Gamma \rightarrow P}{\Gamma \rightarrow Q} \quad (\Rightarrow\text{-}elim)$$

Redundant Proofs
Proof Normalization

$$\dfrac{\dfrac{\dfrac{((R \Rightarrow R) \Rightarrow Q)^x \qquad (R \Rightarrow R)^y}{Q}}{((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q}\,x}{(R \Rightarrow R) \Rightarrow (((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q)}\,y \qquad \dfrac{\dfrac{R^z}{R}}{R \Rightarrow R}\,z}{((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q}$$

Redundant Proofs
Proof Normalization

- When an elimination step immediately follows an introduction step, a proof can be normalized (simplified)

$$\dfrac{\dfrac{\dfrac{\dfrac{((R \Rightarrow R) \Rightarrow Q)^x \qquad (R \Rightarrow R)^y}{Q}}{((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q}\ x}{(R \Rightarrow R) \Rightarrow (((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q)}\ y \qquad \dfrac{\dfrac{R^z}{R}}{R \Rightarrow R}\ z}{((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q}$$

Proof Normalization

- A simpler (normalized) proof:

$$
\dfrac{\dfrac{((R \Rightarrow R) \Rightarrow Q)^x \qquad \dfrac{\dfrac{R^z}{R}}{R \Rightarrow R}\, z}{Q}}{((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q}\, x
$$

Where is that simpler proof?

Pointing at a bad proof!

Normalization and Strong Normalization of Proofs

Normalization and Strong Normalization of Proofs

- In the sixties, Dag Prawitz gave reduction rules.

Normalization and Strong Normalization of Proofs

- In the sixties, Dag Prawitz gave reduction rules.
- He proved that every proof can be reduced to a normal form (normalization).

Normalization and Strong Normalization of Proofs

- In the sixties, Dag Prawitz gave reduction rules.
- He proved that every proof can be reduced to a normal form (normalization).
- In 1971, he proved that every reduction sequence terminates (strong normalization) and that every proof has a unique normal form.

Propositions as types and proofs as simply-typed lambda terms

$$\Gamma, x : P \rightarrow x : P$$

$$\frac{\Gamma, x : P \rightarrow M : Q}{\Gamma \rightarrow \lambda x : P \cdot M : P \Rightarrow Q} \quad (\Rightarrow\text{-}intro)$$

$$\frac{\Gamma \rightarrow M : P \Rightarrow Q \qquad \Gamma \rightarrow N : P}{\Gamma \rightarrow M N : Q} \quad (\Rightarrow\text{-}elim)$$

The Curry-Howard Isomorphism

The Curry-Howard Isomorphism

- Howard (1969) observed that proofs can be represented as terms of the simply-typed lambda-calculus (Church).

The Curry-Howard Isomorphism

- Howard (1969) observed that proofs can be represented as terms of the simply-typed lambda-calculus (Church).
- Propositions can be viewed as types.

The Curry-Howard Isomorphism

- Howard (1969) observed that proofs can be represented as terms of the simply-typed lambda-calculus (Church).
- Propositions can be viewed as types.
- Proof normalization corresponds to lambda-conversion.

The Curry-Howard Isomorphism

- Howard (1969) observed that proofs can be represented as terms of the simply-typed lambda-calculus (Church).
- Propositions can be viewed as types.
- Proof normalization corresponds to lambda-conversion.

The Curry-Howard Isomorphism

- Howard (1969) observed that proofs can be represented as terms of the simply-typed lambda-calculus (Church).
- Propositions can be viewed as types.
- Proof normalization corresponds to lambda-conversion.
- Strong normalization (SN) in the typed lambda-calculus implies SN of proofs.

The Curry-Howard Isomorphism

- Howard (1969) observed that proofs can be represented as terms of the simply-typed lambda-calculus (Church).
- Propositions can be viewed as types.
- Proof normalization corresponds to lambda-conversion.

$$(\lambda x\colon \sigma \cdot M)N \longrightarrow_{\beta} M[N/x]$$

- Strong normalization (SN) in the typed lambda-calculus implies SN of proofs.

Adding the connectives and, or, not

- To deal with negation, we introduce falsity (absurdum), the proposition always false:

$$\perp$$

- We view $\neg P$, the negation of P, as an abbreviation for $P \Rightarrow \perp$

Rules for and

The $\wedge$-*introduction rule*:

$$\frac{\dfrac{\Gamma}{P} \qquad \dfrac{\Delta}{Q}}{P \wedge Q}$$

The $\wedge$-*elimination rule*:

$$\frac{\dfrac{\Gamma}{P \wedge Q}}{P} \qquad \frac{\dfrac{\Gamma}{P \wedge Q}}{Q}$$

Rules for or

The $\vee$-*introduction rule*:

$$\dfrac{\dfrac{\Gamma}{P}}{P \vee Q} \qquad\qquad \dfrac{\dfrac{\Gamma}{Q}}{P \vee Q}$$

The $\vee$-*elimination rule*:

$$\dfrac{\dfrac{\Gamma}{P \vee Q} \qquad \dfrac{\Delta, P^x}{R} \qquad \dfrac{\Lambda, Q^y}{R}}{R}\ x,y$$

Rules for negation

The $\neg$-*introduction rule*:

$$\cfrac{\cfrac{\Gamma, P^x}{\bot}}{\neg P}\ x$$

The $\neg$-*elimination rule*:

$$\cfrac{\cfrac{\Gamma}{\neg P} \qquad \cfrac{\Delta}{P}}{\bot}$$

The Quantifier Rules

$\forall$-*introduction*:

$$\frac{\dfrac{\Gamma}{P[u/t]}}{\forall t P}$$

Here, u must be a variable that does not occur free in any of the propositions in Γ or in $\forall tP$; the notation $P[u/t]$ stands for the result of substituting u for all free occurrences of t in P.

$\forall$-*elimination*:

$$\frac{\dfrac{\Gamma}{\forall t P}}{P[\tau/t]}$$

Here τ is an arbitrary term and it is assumed that bound variables in P have been renamed so that none of the variables in τ are captured after substitution.

The Quantifier Rules

$\exists$-*introduction*:

$$\dfrac{\dfrac{\Gamma}{P[\tau/t]}}{\exists t P}$$

As in $\forall$-elimination, τ is an arbitrary term and the same proviso on bound variables in P applies.

$\exists$-*elimination*:

$$\dfrac{\dfrac{\Gamma}{\exists t P} \qquad \dfrac{\Delta, P[u/t]^x}{C}}{C}\ x$$

Here, u must be a variable that does not occur free in any of the propositions in Δ, $\exists t P$, or C, and all premises $P[u/t]$ labeled x are discharged.

The ``Controversial ” Rules

The $\perp$-*elimination rule*:

$$\frac{\frac{\Gamma}{\perp}}{P}$$

The *proof-by-contradiction rule* (also known as *reductio ad absurdum rule*, for short *RAA*):

$$\frac{\frac{\Gamma, \neg P^x}{\perp}}{P}\, x$$

Problems With Negation

$\bot$-elimination

$\neg\neg P \Rightarrow P$ $\neg P \lor P$

Problems With Negation

- The $\perp$-elimination rule is not so bad.

$$\neg\neg P \Rightarrow P \qquad\qquad \neg P \vee P$$

Problems With Negation

- The $\bot$-elimination rule is not so bad.
- It says that once we have reached an absurdity, then everything goes!

$$\neg\neg P \Rightarrow P \qquad\qquad \neg P \vee P$$

Problems With Negation

- The $\bot$-elimination rule is not so bad.
- It says that once we have reached an absurdity, then everything goes!
- RAA is worse! I allows us to prove double negation elimination and the law of the excluded middle:

$$\neg\neg P \Rightarrow P \qquad\qquad \neg P \vee P$$

Problems With Negation

- The $\bot$-elimination rule is not so bad.
- It says that once we have reached an absurdity, then everything goes!
- RAA is worse! I allows us to prove double negation elimination and the law of the excluded middle:
- $\neg\neg P \Rightarrow P$ $\neg P \vee P$

Problems With Negation

- The $\perp$-elimination rule is not so bad.
- It says that once we have reached an absurdity, then everything goes!
- RAA is worse! I allows us to prove double negation elimination and the law of the excluded middle:
- $\neg\neg P \Rightarrow P$ $\qquad\qquad$ $\neg P \vee P$
- Constructively, these are problematic!

Lack of Constructivity

- The provability of $\neg\neg P \Rightarrow P$ and $\neg P \vee P$ is equivalent to RAA.
- RAA allows proving disjunctions (and existential statements) that may not be constructive; this means that if $A \vee B$ is provable, in general, it may not be possible to give a proof of A or a proof of B
- This lack of constructivity of classical logic led Brouwer to invent intuitionistic logic

That's too abstract, give me something concrete!

A non-constructive proof

- Claim: There exist two reals numbers, a, b, both irrational, such that a^b is rational.
- Proof: We know that $\sqrt{2}$ is irrational. Either
- (1) $\sqrt{2}^{\sqrt{2}}$ is rational; $a = b = \sqrt{2}$, or
- (2) $\sqrt{2}^{\sqrt{2}}$ is irrational; $a = \sqrt{2}^{\sqrt{2}}, \ b = \sqrt{2}$
- In (2), we use $(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = 2$
- Using the law of the excluded middle, our claim is proved! But, what is $\sqrt{2}^{\sqrt{2}}$?

Non-constructive Proofs

- The previous proof is non-constructive.
- It shows that a and b must exist but it does not produce an explicit solution.
- This proof gives no information as to the irrationality of $\sqrt{2}^{\sqrt{2}}$
- In fact, $\sqrt{2}^{\sqrt{2}}$ is irrational, but this is very hard to prove!
- A ``better” solution: $a = \sqrt{2},\ b = \log_2 9$

Existence proofs are often non-constructive

- Fixed-points Theorems often only assert the existence of a fixed point but provide no method for computing them.
- For example, Brouwer's Fixed Point Theorem.
- That's too bad, this theorem is used in the proof of the Nash Equilibrium Theorem!

Intuitionism (Brouwer, Heyting)

Intuitionism (Brouwer, Heyting)

- L E J Brouwer (1881-1966)

Intuitionism (Brouwer, Heyting)

- L E J Brouwer (1881-1966)
- Founder of intuitionism (1907)

Intuitionism (Brouwer, Heyting)

- L E J Brouwer (1881-1966)
- Founder of intuitionism (1907)
- Also important work in topology

A. Heyting

A. Heyting

- Arend Heyting (1898-1980)

A. Heyting

- Arend Heyting (1898-1980)
- Heyting algebras (semantics for intuitionistic logic)

Intuitionistic Logic

- In intuitionistic logic, it is forbidden to use the proof by contradiction rule (RAA)
- As a consequence, $\neg\neg P$ no longer implies P and $\neg P \vee P$ is no longer provable (in general)
- The connectives, and, or, implication and negation are independent
- No de Morgan laws

Intuitionistic Logic

- Fewer propositions are provable (than in classical logic) but proofs are more constructive.
- If a disjunction, $P \vee Q$, is provable, then a proof of P or a proof of Q can be found.
- Similarly, if $\exists t P$ is provable, then there is a term, τ, such that $P[\tau/t]$ is provable.
- However, the complexity of proof search is higher.

Intuitionistic Logic and Typed lambda-Calculi

Intuitionistic Logic and Typed lambda-Calculi

- Proofs in intuitionistic logic can be represented as certain kinds of lambda-terms.

Intuitionistic Logic and Typed lambda-Calculi

- Proofs in intuitionistic logic can be represented as certain kinds of lambda-terms.
- We now have conjunctive, disjunctive, universal and existential types.

Intuitionistic Logic and Typed lambda-Calculi

- Proofs in intuitionistic logic can be represented as certain kinds of lambda-terms.
- We now have conjunctive, disjunctive, universal and existential types.
- Falsity can be viewed as an ``error type''

Intuitionistic Logic and Typed lambda-Calculi

- Proofs in intuitionistic logic can be represented as certain kinds of lambda-terms.
- We now have conjunctive, disjunctive, universal and existential types.
- Falsity can be viewed as an ``error type”
- Strong Normalization still holds, but some subtleties with disjunctive and existential types (permutative reductions)

Higher-order Intuitionistic Logic

Higher-order Intuitionistic Logic

- We allow quantification over functions.

Higher-order Intuitionistic Logic

- We allow quantification over functions.
- The corresponding lambda-calculus is a polymorphic lambda calculus (first invented by J.Y. Girard, systems F and F-omega, 1971)

Higher-order Intuitionistic Logic

- We allow quantification over functions.
- The corresponding lambda-calculus is a polymorphic lambda calculus (first invented by J.Y. Girard, systems F and F-omega, 1971)
- System F was independently discovered by J. Reynolds (1974) for very different reasons.

Higher-order Intuitionistic Logic

- We allow quantification over functions.
- The corresponding lambda-calculus is a polymorphic lambda calculus (first invented by J.Y. Girard, systems F and F-omega, 1971)
- System F was independently discovered by J. Reynolds (1974) for very different reasons.
- Later, even richer typed calculi, the theory of construction (Coquand, Huet)

Proof Search

Proof Search

- Some rules (or-elim, exists-elim) violate the subformula property

Proof Search

- Some rules (or-elim, exists-elim) violate the subformula property
- This makes searching for proofs very expansive

Proof Search

- Some rules (or-elim, exists-elim) violate the subformula property
- This makes searching for proofs very expansive
- Natural deduction systems are not well suited for (automated) proof search

Proof Search

- Some rules (or-elim, exists-elim) violate the subformula property
- This makes searching for proofs very expansive
- Natural deduction systems are not well suited for (automated) proof search
- Gentzen sequent calculi are much better suited for proof search.

Pelikans Proof Searching

Proof Search (Sequent Calculi)

- A Gentzen sequent is a pair of sets of formulae, $\Gamma \rightarrow \Delta$, where

$$\Gamma = \{P_1, \ldots, P_m\} \qquad \Delta = \{Q_1, \ldots, Q_n\}$$

- The intuitive idea is that if all the propositions in Γ hold, then some proposition in Δ should hold.
- The rules of a Gentzen system break the formulae P_i and Q_j into subformulae that may end up on the other side of the arrow

Proof Search (Sequent Calculi)

- In intuitionistic logic, Δ has at most one formula
- In classical propositional logic, every search strategy terminates.
- In intuitionistic propositional logic, there is a search strategy that always terminates.
- In first-order logic (classical, intuitionistic), there is no general search procedure that always terminates (Church's Theorem).

Triumph Proof Searching

What about Semantics?

What about Semantics?

- For classical propositional logic: truth values semantics ({true, false}).

What about Semantics?

- For classical propositional logic: truth values semantics ({true, false}).
- For intuitionistic propositional logic: Heyting algebras, Kripke models.

What about Semantics?

- For classical propositional logic: truth values semantics ({true, false}).
- For intuitionistic propositional logic: Heyting algebras, Kripke models.
- For classical first-order logic: first-order structures (Tarskian semantics).

What about Semantics?

- For classical propositional logic: truth values semantics ({true, false}).
- For intuitionistic propositional logic: Heyting algebras, Kripke models.
- For classical first-order logic: first-order structures (Tarskian semantics).
- For intuitionistic first-order logic: Kripke models.

Soundness and Completeness

Soundness and Completeness

- Soundness: Every provable formula is valid (has the value true for all interpretations).

Soundness and Completeness

- Soundness: Every provable formula is valid (has the value true for all interpretations).
- A proof system must be sound or else it is garbage!

Soundness and Completeness

- Soundness: Every provable formula is valid (has the value true for all interpretations).
- A proof system must be sound or else it is garbage!
- Completeness: Every valid formula is provable.

Soundness and Completeness

- Soundness: Every provable formula is valid (has the value true for all interpretations).
- A proof system must be sound or else it is garbage!
- Completeness: Every valid formula is provable.
- Completeness is desirable but not always possible.

Completeness: Good News

Completeness: Good News

- The systems I presented are all sound and complete.

Completeness: Good News

- The systems I presented are all sound and complete.
- Godel (completeness theorem for classical logic)

Completeness: Good News

- The systems I presented are all sound and complete.
- Godel (completeness theorem for classical logic)
- Kripke (completeness theorem for intuitionistic logic)

Completeness: Good News

- The systems I presented are all sound and complete.
- Godel (completeness theorem for classical logic)
- Kripke (completeness theorem for intuitionistic logic)
- Classical Propositional validity: decidable.

Completeness: Good News

- The systems I presented are all sound and complete.
- Godel (completeness theorem for classical logic)
- Kripke (completeness theorem for intuitionistic logic)
- Classical Propositional validity: decidable.
- Intuitionistic Propositional validity: decidable

Completeness: Bad News!

Completeness: Bad News!

- Complexity of classical prop. validity: co-NP complete (Cook, Karp, 1970)

Completeness: Bad News!

- Complexity of classical prop. validity: co-NP complete (Cook, Karp, 1970)
- Complexity of intuitionistic prop. validity: P-space complete! (Statman, 1979)

Completeness: Bad News!

- Complexity of classical prop. validity: co-NP complete (Cook, Karp, 1970)
- Complexity of intuitionistic prop. validity: P-space complete! (Statman, 1979)
- The decision problem (validity problem) for first-order (classical) logic is undecidable (Church, 1936)

Completeness: Bad News!

- Complexity of classical prop. validity: co-NP complete (Cook, Karp, 1970)
- Complexity of intuitionistic prop. validity: P-space complete! (Statman, 1979)
- The decision problem (validity problem) for first-order (classical) logic is undecidable (Church, 1936)
- Decision problem for intuitionistic logic also undecidable (double negation translation)

Kurt Godel (1906-1978)
(Right: with A. Einstein)

Alonzo Church
(1903-1995)

Proof Search in Classical Logic

Proof Search in Classical Logic

- Herbrand's idea: Reduce the provability of a first-order formula to the provability of a quantifier-free conjunction of substitution instances of this formula.

Proof Search in Classical Logic

- Herbrand's idea: Reduce the provability of a first-order formula to the provability of a quantifier-free conjunction of substitution instances of this formula.
- Normal forms become crucial: conjunctive normal form (cnf), negation normal form (nnf)

Proof Search in Classical Logic

- Herbrand's idea: Reduce the provability of a first-order formula to the provability of a quantifier-free conjunction of substitution instances of this formula.
- Normal forms become crucial: conjunctive normal form (cnf), negation normal form (nnf)
- Nice formulation of Herbrand's Theorem for formulae in nnf due to Peter Andrews

Substitutions, Unification

- Roughly speaking, compound instances are obtained by recursively substituting terms for variables in subformulae.
- It turns out that the crux of the method is to find substitutions so that

$$\sigma(P_i) = \sigma(P_j)$$

- where P_i, P_j are atomic formulae occurring with opposite signs

Unification Procedures

Unification Procedures

- Such substitutions are called unifiers

Unification Procedures

- Such substitutions are called unifiers
- For efficiency reasons, it is important to find most general unifiers (mgu's)

Unification Procedures

- Such substitutions are called unifiers
- For efficiency reasons, it is important to find most general unifiers (mgu's)
- mgu's always exist. There are efficient algorithms for finding them (Martelli-Montanari, Paterson and Wegman)

Unification Procedures

- Such substitutions are called unifiers
- For efficiency reasons, it is important to find most general unifiers (mgu's)
- mgu's always exist. There are efficient algorithms for finding them (Martelli-Montanari, Paterson and Wegman)
- Higher-order unification is also of great interest, but undecidable in general!

Some Theorem Provers and Proof Assistants

- Isabelle
- COQ (Benjamin Pierce is writing two books that make use of COQ)
- TPS
- NUPRL
- PVS
- Agda
- Twelf

Other Logics?

Other Logics?

- One will note that in a deduction (natural or Gentzen sequent style), the same premise can be used as many times as needed.

Other Logics?

- One will note that in a deduction (natural or Gentzen sequent style), the same premise can be used as many times as needed.
- Girard (and Lambeck earlier) had the idea to restrict the use of premises (charge for multiple use).

Other Logics?

- One will note that in a deduction (natural or Gentzen sequent style), the same premise can be used as many times as needed.
- Girard (and Lambeck earlier) had the idea to restrict the use of premises (charge for multiple use).
- This leads to logics where the connectives have a double identity: additive or multiplicative.

Finer Logics: Linear Logic, ...

Finer Logics: Linear Logic, ...

- linear logic, invented by Girard, achieves much finer control over the use of premises.

Finer Logics: Linear Logic, ...

- linear logic, invented by Girard, achieves much finer control over the use of premises.
- The notion of proof becomes more general: proof nets (certain types of graphs)

Finer Logics: Linear Logic, ...

- linear logic, invented by Girard, achieves much finer control over the use of premises.
- The notion of proof becomes more general: proof nets (certain types of graphs)
- linear logic can be viewed as an attempt to deal with resources and parallelism

Finer Logics: Linear Logic, ...

- linear logic, invented by Girard, achieves much finer control over the use of premises.
- The notion of proof becomes more general: proof nets (certain types of graphs)
- linear logic can be viewed as an attempt to deal with resources and parallelism
- Negation is an involution

Special Purpose Logics: Temporal, ...

Special Purpose Logics: Temporal, ...

- From a practical point of view, it is very fruitful to design logics with intented semantics, such as time, concurrency, ...

Special Purpose Logics: Temporal, ...

- From a practical point of view, it is very fruitful to design logics with intented semantics, such as time, concurrency, ...
- Temporal logic deals with time (A. Pnueli)

Special Purpose Logics: Temporal, ...

- From a practical point of view, it is very fruitful to design logics with intented semantics, such as time, concurrency, ...
- Temporal logic deals with time (A. Pnueli)
- Process logic (Manna, Pnueli)

Special Purpose Logics: Temporal, ...

- From a practical point of view, it is very fruitful to design logics with intented semantics, such as time, concurrency, ...
- Temporal logic deals with time (A. Pnueli)
- Process logic (Manna, Pnueli)
- Dynamic logic (Harel, Pratt)

Special Purpose Logics: Temporal, ...

- From a practical point of view, it is very fruitful to design logics with intented semantics, such as time, concurrency, ...
- Temporal logic deals with time (A. Pnueli)
- Process logic (Manna, Pnueli)
- Dynamic logic (Harel, Pratt)
- The world of logic is alive and well!

Searching for that proof!