An introduction to Kleene realizability

Alexandre Miquel

UNIVERSIDAD DE LA REPUBLICA URUGUAY
FACULTAD DE INGENIERIA
EQUIPO DE LÓGICA UDELAR

October 13th & 20th, 2021

A disjunction without alternative

Theorem

At least one of the two numbers $e+\pi$ and $e\pi$ is transcendental

Proof

Reductio ad absurdum: Suppose $S = e+\pi$ and $P = e\pi$ are algebraic.
Then e, π are solutions of the polynomial with algebraic coefficients

$$X^2 - SX + P = 0$$

Hence e and π are algebraic. Contradiction.

- Proof does not say which of $e+\pi$ and/or $e\pi$ is transcendental
  (The problem of the transcendence of $e+\pi$ and $e\pi$ is still open.)
- Non constructivity comes from the use of reductio ad absurdum

An existence without a witness

Theorem

There are two irrational numbers a and b such that a^b is rational.

Proof

Either $\sqrt{2}^{\sqrt{2}} \in \mathbb{Q}$ or $\sqrt{2}^{\sqrt{2}} \notin \mathbb{Q}$, by excluded middle. We reason by cases:

- If $\sqrt{2}^{\sqrt{2}} \in \mathbb{Q}$, take $a = b = \sqrt{2} \notin \mathbb{Q}$.
- If $\sqrt{2}^{\sqrt{2}} \notin \mathbb{Q}$, take $a = \sqrt{2}^{\sqrt{2}} \notin \mathbb{Q}$ and $b = \sqrt{2} \notin \mathbb{Q}$, since:

$$a^b = \left(\sqrt{2}^{\sqrt{2}}\right)^{\sqrt{2}} = (\sqrt{2})^{(\sqrt{2}\times\sqrt{2})} = (\sqrt{2})^2 = 2 \in \mathbb{Q}$$

- Proof does not say which of $(\sqrt{2}, \sqrt{2})$ or $\left(\sqrt{2}^{\sqrt{2}}, \sqrt{2}\right)$ is solution
- Non constructivity comes from the use of excluded middle
- But there are constructive proofs, e.g.: $(a, b) = \left(\sqrt{2},\ 2\log_2 3\right)$

The first non constructive proof

- Historically, excluded middle and reductio ad absurdum are known since antiquity (Aristotle). But they were never used in an essential way until the end of the 19th century. Example:

Theorem

There exist transcendental numbers

Constructive proof, by Liouville 1844

The number $a = \sum_{n=1}^{\infty} \frac{1}{10^{n!}} = 0.110001000000\cdots$ is transcendental.

Non constructive proof, by Cantor 1874

Since $\mathbb{Z}[X]$ is denumerable, the set $\mathbb{A}$ of algebraic numbers is denumerable. But $\mathbb{R} \sim \mathfrak{P}(\mathbb{N})$ is not. Hence $\mathbb{R} \setminus \mathbb{A}$ is not empty and even uncountable.

Plan

Plan

Brouwer's intuitionism

Luitzen Egbertus Jan **Brouwer** (1881–1966)

1908: *The untrustworthiness of the principles of logic*

- Rejection of non constructive principles such as:
  - The law of excluded-middle ($A \vee \neg A$)
  - Reductio ad absurdum (deduce A from the absurdity of $\neg A$)
  - The Axiom of Choice, actually: only its strongest forms (Zorn)
- Principles of intuitionism:
  - Philosophy of the creative subject
  - Each mathematical object is a construction of the mind.
    Proofs themselves are constructions (methods, rules...)
  - Rejection of Hilbert's formalism (no formal rules!)

Brouwer also made fundamental contributions to classical topology (fixed point theorem, invariance of the domain)... only to be accepted in the academia

Intuitionistic Logic (LJ)

Although Brouwer was deeply opposed to formalism, the rules of Intuitionistic Logic (LJ) were formalized by his student Arend **Heyting** (1898–1990)

1930: *The formal rules of intuitionistic logic*

1956: *Intuitionism. An introduction*

Intuitively:

- Constructions $A \wedge B$ and $\forall x\, A(x)$ keep their usual meaning, but constructions $A \vee B$ and $\exists x\, A(x)$ get a stronger meaning:
  - A proof of $A \vee B$ should implicitly decide which of A or B holds
  - A proof of $\exists x\, A(x)$ should implicitly construct x
- Implication $A \Rightarrow B$ has now a procedural meaning (cf later) and negation $\neg A$ (defined as $A \Rightarrow \bot$) is no more involutive

Technically: LJ $\subset$ LK (LK $=$ classical logic)

Intuitionistic logic: what we keep / what we lose

- We keep the implications...

$$A \Rightarrow \neg\neg A \qquad \text{(Double negation)}$$
$$(A \Rightarrow B) \Rightarrow (\neg B \Rightarrow \neg A) \qquad \text{(Contraposition)}$$
$$(\neg A \vee B) \Rightarrow (A \Rightarrow B) \qquad \text{(Material implication)}$$
$$\neg A \Leftrightarrow \neg\neg\neg A \qquad \text{(Triple negation)}$$

  but converse implications are lost (but the last)

- De Morgan laws:

$$\neg(A \vee B) \Leftrightarrow \neg A \wedge \neg B \qquad \neg(A \wedge B) \Leftarrow \neg A \vee \neg B$$
$$\neg(\exists x\ A(x)) \Leftrightarrow \forall x\ \neg A(x) \qquad \neg(\forall x\ A(x)) \Leftarrow \exists x\ \neg A(x)$$

- **Beware!** Do not confound the two rules:

$$\frac{A \vdash \bot}{\vdash \neg A} \quad \left(\begin{array}{l}\text{introduction rule of}\\ \text{negation, accepted,}\\ \text{cf proof of } \sqrt{2} \notin \mathbb{Q}\end{array}\right) \quad \text{and} \quad \frac{\neg A \vdash \bot}{\vdash A} \quad \left(\begin{array}{c}\text{Reductio ad}\\ \text{absurdum,}\\ \text{rejected}\end{array}\right)$$

Intuitionistic mathematics: what we keep / what we lose

In Algebra:

- We keep all basic algebra, and most of abstract algebra
- The theory of orders is almost entirely kept
- The same for combinatorics

In Topology:

- General topology needs to be entirely reformulated: topology without points, formal spaces

In Analysis:

- $\mathbb{R}$ still exists, but it is no more unique! (Depends on the construction)
- Functions on compact sets do not reach their maximum
- We can reformulate Borel/Lebesgue measure & integral, using the suitable construction of $\mathbb{R}$ [Coquand'02]

A note on decidability

- Intuitionistic mathematicians have nothing against statements of the form $A \vee \neg A$. They just need to be proved... constructively
  - LJ $\vdash$ $(\forall x, y \in \mathbb{N})(x = y \vee x \neq y)$ (equality is decidable on $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$)
  - LJ $\nvdash$ $(\forall x, y \in \mathbb{R})(x = y \vee x \neq y)$ (equality is undecidable on $\mathbb{R}$, $\mathbb{C}$)
- More generally, the formula $(\forall \vec{x} \in S)\,(A(\vec{x}) \vee \neg A(\vec{x}))$
  is intended to mean: "Predicate/relation A is decidable on S"
- This intuitionistic notion of 'decidability' can be formally related to the mathematical (C.S.) notion of decidability using realizability
- **Variant:** Trichotomy
  - LJ $\vdash$ $(\forall x, y \in \mathbb{N})(x < y \vee x = y \vee x > y)$
  - LJ $\nvdash$ $(\forall x, y \in \mathbb{R})(x < y \vee x = y \vee x > y)$, but:
  - LJ $\vdash$ $(\forall x, y \in \mathbb{R})(x \neq y \Rightarrow x < y \vee x > y)$

The jungle of intuitionistic theories

- At the lowest levels of mathematics, intuitionism is well-defined:
  - LJ: Intuitionistic (predicate) logic
  - HA: Heyting Arithmetic (= intuitionistic arithmetic)
  - + some well-known extensions of HA (e.g. Markov principle)
- But as we go higher, definition is less clear. Two trends:
- **Predicative theories:** ("Swedish school")
  - Bishop's constructive analysis
  - Martin-Löf type theories (MLTT)
  - Aczel's constructive set theory (CZF)
- **Impredicative theories:** ("French school")
  - Girard's system F
  - Coquand-Huet's calculus of constructions
  - The Coq proof assistant
  - Intuitionistic Zermelo Fraenkel (IZF_R, IZF_C) [Myhill-Friedman 1973]

Brouwer's contribution to classical mathematics

Brouwer also made fundamental contributions to classical topology, especially in the theory of topological manifolds:

Theorem (Fixed point Theorem)

Any continuous function $f : D^n \to D^n$ has a fixed point $(D^n = \text{unit ball of } \mathbb{R}^n)$

Theorem (Invariance of the domain)

Let $U \subseteq \mathbb{R}^n$ be an open set, and $f : U \to \mathbb{R}^n$ continuous and injective.
Then $f(U)$ is open, and the function f is open.

Corollary (Topological invariance of dimension)

Let $U \subseteq \mathbb{R}^n$ and $V \subseteq \mathbb{R}^m$ be nonempty open sets.
If U and V are homeomorphic, then $n = m$.

... but these results use classical reasoning in an essential way,
and were never regarded as valid by Brouwer

What does it mean to be constructive for a theory? (1/2)

- There is no fixed criterion for a theory $\mathscr{T}$ to be constructive, but a mix of syntactical, semantical and philosophical criteria
- But it should fulfill at least the following 4 criteria:

(1) $\mathscr{T}$ should be recursive. Which means that the sets of axioms, derivations and theorems of $\mathscr{T}$ are all recursively enumerable

Note: This is already the case for standard classical theories: PA, ZF, ZFC, etc.

(2) $\mathscr{T}$ should be consistent: $\mathscr{T} \nvdash \bot$

(3) $\mathscr{T}$ should satisfy the disjunction property:

$$\text{If} \quad \mathscr{T} \vdash A \vee B, \quad \text{then} \quad \mathscr{T} \vdash A \quad \text{or} \quad \mathscr{T} \vdash B$$

(where A, B are closed formulas)

(4) $\mathscr{T}$ should satisfy the numeric existence property:

$$\text{If} \quad \mathscr{T} \vdash (\exists x \in \mathbb{N})\, A(x), \quad \text{then} \quad \mathscr{T} \vdash A(n) \quad \text{for some } n \in \mathbb{N}$$

(where $A(x)$ only depends on x)

What does it mean to be constructive for a theory? (2/2)

- In most cases, we also require that:

  (5) $\mathscr{T}$ should satisfy the existence property (or witness property):

If $\mathscr{T} \vdash \exists x\, A(x)$, then $\mathscr{T} \vdash A(t)$ for some closed term t

(where $A(x)$ only depends on x)

Note: Needs to be adapted when the language of $\mathscr{T}$ has no closed term (for instance: set theory)

Theorem (Non constructivity of classical theories)

If a classical theory is recursive, consistent and contains Q, then it fulfills none of the disjunction and numeric existence properties

Note: Q = Robinson Arithmetic ($\subset$ PA), that is: the (finite) fragment of PA where the induction scheme is replaced by the (much weaker) axiom $\forall x\,(x = 0 \lor \exists y\,(x = s(y)))$

Proof. From the hypotheses, Gödel's 1st incompleteness theorem applies, so we can pick a closed formula G such that $\mathscr{T} \nvdash G$ and $\mathscr{T} \nvdash \neg G$. We conclude noticing that:

$$\mathscr{T} \vdash G \lor \neg G \qquad \text{and} \qquad \mathscr{T} \vdash (\exists x \in \mathbb{N})\,((x = 1 \land G) \lor (x = 0 \land \neg G))$$

Why using LJ does not ensure constructivity (1/2)

- Constructivity is a semantical (and philosophical) criterion, that cannot be simply ensured by the use of intuitionistic logic (LJ)
- Indeed, some awkward axiomatizations in LJ may imply the excluded middle, and thus lead to non constructive theories. Some examples:
- **In intuitionistic arithmetic (HA):**
  - The axiom of well-ordering

    $$(\forall S \subseteq \mathbb{N})\,[\exists x\,(x \in S) \;\Rightarrow\; (\exists x \in S)(\forall y \in S)\,x \leq y]$$

    implies the excluded middle; it is not constructive. In HA, induction (which is constructive) does not imply well-ordering

Why using LJ does not ensure constructivity (2/3)

- **In constructive analysis:** [Bishop 1967]
  - The axiom of trichotomy

    $$(\forall x, y \in \mathbb{R})\,(x < y \lor x = y \lor x > y)$$

    is not constructive. It has to be replaced by the axiom

    $$(\forall x, y \in \mathbb{R})\,(x \neq y \Rightarrow x < y \lor x > y)$$

    which is classically equivalent
  - The axiom of completeness

    > Each inhabited subset of $\mathbb{R}$ that has an upper bound in $\mathbb{R}$ has a least upper bound in $\mathbb{R}$

    implies excluded middle. It has to be restricted to the inhabited subsets $S \subseteq \mathbb{R}$ that are order located above, i.e., such that:

    For all $a < b$, either $(\forall x \in S)\,(x \leq b)$ or $(\exists x \in S)\,(x \geq a)$

Why using LJ does not ensure constructivity (3/3)

- **In Intuitionistic Set Theory:**
  - The classical formulation of the Axiom of Regularity (or Foundation)

    $$\forall x\,(x \neq \varnothing \Rightarrow (\exists y \in x)(y \cap x \neq 0))$$

    implies excluded middle. It has to be replaced by the axiom scheme

    $$\forall x\,((\forall y \in x)\,A(y) \;\Rightarrow\; A(x)) \;\Rightarrow\; \forall x\,A(x)$$

    known as set induction, that is classically equivalent
  - The set-theoretic Axiom of Choice (Zorn, Zermelo, etc.) implies excluded middle [Diaconescu 1975]

- In all cases, the constructivity of a given intuitionistic theory $\mathscr{T}$ is justified by realizability techniques... (for criteria (2)–(5))
  - ... either directly (realizability model)
  - ... either indirectly (type system + normalization)

Plan

The language of Arithmetic

First-order terms and formulas

FO-terms	e, e_1	$::=$	$x \mid f(e_1, \ldots, e_k) \qquad (f \text{ of arity } k)$
Formulas	A, B	$::=$	$e_1 = e_2 \mid \top \mid \bot \mid A \Rightarrow B$
			$\mid A \wedge B \mid A \vee B \mid \forall x\, A \mid \exists x\, A$

- We assume given one k-ary function symbol f for each primitive recursive function of arity k: $0, s, +, -, \times, \uparrow$, etc.
- Only one (binary) predicate symbol: $=$ (equality)
- Macros: $\neg A \;:=\; A \Rightarrow \bot, \quad A \Leftrightarrow B \;:=\; (A \Rightarrow B) \wedge (B \Rightarrow A)$
- **Syntactic worship:** Free & bound variables. Work up to α-conversion. Set of free variables: $FV(e)$, $FV(A)$. Substitution: $e[x := e_0]$, $A[x := e_0]$.

Choice of a deduction system

- There are many equivalent ways to present the deduction rules of intuitionistic (or classical) predicate logic:
  1. In the style of Hilbert (only formulas, no sequents)
  2. In the style of Gentzen (left & right rules)
  3. In the style of Natural Deduction (with or without sequents)

  Since these systems define the very same class of provable formulas[1] (for a given logic, LJ or LK), choice is just a matter of convenience

- Systems only based on formulas (Hilbert's, N.D. without sequents) are easier to define, but much more difficult to manipulate

- In what follows, we shall systematically use sequents

[1] In sequent-based systems, formulas are identified with sequents of the form $\vdash A$, that is: with sequents with 0 hypothesis (lhs) and 1 thesis (rhs)

Sequents

Definition (Sequent) [Gentzen 1934]

A sequent is a pair of finite lists of formulas written

$$A_1, \ldots, A_n \vdash B_1, \ldots, B_m \qquad (n, m \geq 0)$$

- $A_1, \ldots, A_n$ are the hypotheses (which form the antecedent)
- $B_1, \ldots, B_m$ are the theses (which form the consequent)
- $\vdash$ is the entailment symbol (that reads: 'entails')

Note: Some authors use finite multisets (of formulas) rather than finite lists, since the order is irrelevant, both in the antecedent and in the consequent

- Sequents are usually written $\Gamma \vdash \Delta$ (Γ, Δ finite lists of formulas)
- Intuitive meaning: $\bigwedge \Gamma \ \Rightarrow \ \bigvee \Delta$
- Empty sequent " $\vdash$ " represents contradiction
- **Syntactic worship:** Notations $FV(\Gamma)$, $\Gamma[x := e]$ extended to finite lists Γ

Rules of inference & systems of deduction

Formulas and sequents can be used as judgments. Each system of deduction is based on a set of judgments $\mathscr{J}$ (= a set of expressions asserting something)

- Given a set of judgments $\mathscr{J}$:

Definition (Rule of inference)

A rule of inference is a pair formed by a finite set of judgments $\{J_1, \ldots, J_n\} \subseteq \mathscr{J}$ and a judgment $J \in \mathscr{J}$, usually written

$$\frac{J_1 \quad \cdots \quad J_n}{J}$$

- $J_1, \ldots, J_n$ are the premises of the rule
- J is the conclusion of the rule

Definition (System of deduction)

A system of deduction is a set of inference rules

Derivable judgments (1/2)

Definition (Derivation)

Let $\mathscr{S}$ be a system of deduction based on some set of judgments $\mathscr{J}$.

1. Derivations (of judgments) in $\mathscr{S}$ are inductively defined as follows:

   If $d_1, \ldots, d_n$ are derivations of $J_1, \ldots, J_n$ in $\mathscr{S}$, respectively, and if $(\{J_1, \ldots, J_n\}, J)$ is a rule of $\mathscr{S}$, then

$$d \;=\; \left\{ \frac{\begin{array}{c}\vdots\; d_1 \\ J_1\end{array} \quad \cdots \quad \begin{array}{c}\vdots\; d_n \\ J_n\end{array}}{J} \right. \qquad \text{is a derivation of } J \text{ in } \mathscr{S}$$

2. A judgment J is derivable in $\mathscr{S}$ when there is a derivation of J in $\mathscr{S}$

- By definition, the set of derivable judgments of $\mathscr{S}$ is the smallest set of judgments that is closed under the rules in $\mathscr{S}$
- One also uses proof/provable for derivation/derivable

Derivable judgments (2/2)

- Two systems of deduction (based on the same set of judgments) are equivalent when the induce the same set of derivable judgments

Definition (Admissible rule)

A rule $R = (\{J_1, \ldots, J_n\}, J)$ is admissible in a system of deduction $\mathscr{S}$ when: $J_1, \ldots, J_n$ derivable in $\mathscr{S}$ implies J derivable in $\mathscr{S}$.

Admissible rules are usually written

$$\frac{J_1 \quad \cdots \quad J_n}{J}$$

- Clearly: R admissible in $\mathscr{S}$ iff $\mathscr{S} \cup \{R\}$ equivalent to $\mathscr{S}$
- **Remark:** In practice, deduction systems are defined as finite sets of schemes of rules (that is: families of rules), that are still called rules. The notion of admissible rule immediately extends to schemes

A remark on implication

In logic, we have (at least) three symbols to represent implication:

- The implication symbol $\Rightarrow$, used in formulas. Represents a potential point for deduction, but not an actual deduction step
- The entailment symbol $\vdash$, used in sequents. Same thing as $\Rightarrow$, but in a sequent, that represents a formula under decomposition:

$$\begin{array}{rl} & A_1, \ldots, A_n \vdash B_1, \ldots, B_m \\ \approx & A_1 \land \cdots \land A_n \Rightarrow B_1 \lor \cdots \lor B_m \end{array}$$

  (So that $\vdash$ is a distinguished implication, closer to a point of deduction)

- The inference rule "$\overline{\qquad\qquad}$", used in rules & derivations. This symbol represents an actual deduction step:

$$\frac{P_1 \quad \cdots \quad P_n}{C} \qquad \left(\begin{array}{l}\text{From } P_1,\ldots,P_n \\ \text{deduce } C\end{array}\right)$$

On the meaning of sequents

- Sequents are not intended to enrich the expressiveness of a logical system; they are only intended to represent a state in a proof, or a formula under decomposition:

$$\Gamma \vdash \Delta \qquad \approx \qquad \bigwedge \Gamma \Rightarrow \bigvee \Delta$$

(With the conventions $\bigwedge \varnothing := \top$ and $\bigvee \varnothing := \bot$)

- **Formally:** In most (if not all[2]) systems in the literature, we have:

$$\Gamma \vdash \Delta \;\text{ derivable} \qquad \text{iff} \qquad \vdash \left(\bigwedge \Gamma \Rightarrow \bigvee \Delta\right) \;\text{ derivable}$$

This equivalence holds, at least:

  - In Gentzen's sequent calculus (LK)
  - In intuitionistic sequent calculus (LJ)
  - In intuitionistic/classical natural deduction (NJ/NK)
  - In Linear Logic (LL), replacing $\wedge$, $\vee$, $\top$, $\bot$, $\Rightarrow$ by $\otimes$, $\parr$, 1, $\bot$, $\multimap$

- **Exercise:** Check it for both systems NJ/NK presented hereafter

[2] The author knows no exception to this rule

Intuitionistic Natural Deduction (NJ)

- Intuitionistic Natural Deduction (NJ) is a deduction system based on asymmetric sequents of the form:

$$A_1, \ldots, A_n \vdash A \qquad \text{or:} \qquad \Gamma \vdash A$$

  These sequents are also called intuitionistic sequents

- Recall that: $\Gamma \vdash A$ has the same meaning as $\bigwedge \Gamma \Rightarrow A$

- System NJ has three kinds of (schemes of) rules:
  - Introduction rules, defining how to prove each connective/quantifier
  - Elimination rules, defining how to use each connective/quantifier
  - The Axiom rule, which is a conservation rule

- The Trimūrti of logic:

Introduction rules	=	Brahma
Elimination rules	=	Shiva
Axiom rule	=	Vishnu

Deduction rules of NJ (1/2)

- Rules for the intuitionistic propositional calculus:

	Introduction	Elimination
(Axiom)	$\dfrac{}{\Gamma \vdash A}\ {\scriptstyle A\in\Gamma}$	
$(\Rightarrow)$	$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$	$\dfrac{\Gamma \vdash A \Rightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B}$
$(\wedge)$	$\dfrac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$	$\dfrac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \qquad \dfrac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}$
$(\vee)$	$\dfrac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \qquad \dfrac{\Gamma \vdash B}{\Gamma \vdash A \vee B}$	$\dfrac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}$
$(\top)$	$\dfrac{}{\Gamma \vdash \top}$	(no elimination rule)
$(\bot)$	(no introduction rule)	$\dfrac{\Gamma \vdash \bot}{\Gamma \vdash A}$

Deduction rules of NJ (2/2)

- Introduction & elimination rules for quantifiers:

$$(\forall)\quad \frac{\Gamma \vdash A}{\Gamma \vdash \forall x\, A}\ {\scriptstyle x \notin FV(\Gamma)} \qquad\Big|\qquad \frac{\Gamma \vdash \forall x\, A}{\Gamma \vdash A[x := e]}$$

$$(\exists)\quad \frac{\Gamma \vdash A[x := e]}{\Gamma \vdash \exists x\, A} \qquad\Big|\qquad \frac{\Gamma \vdash \exists x\, A \qquad \Gamma, A \vdash B}{\Gamma \vdash B}\ {\scriptstyle x \notin FV(\Gamma, B)}$$

- Introduction & elimination rules for equality:

$$(=)\quad \frac{}{\Gamma \vdash e = e} \qquad\Big|\qquad \frac{\Gamma \vdash e_1 = e_2 \qquad \Gamma \vdash A[x := e_1]}{\Gamma \vdash A[x := e_2]}$$

- To get Classical Natural Deduction (NK), just replace

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash A}\ \text{(ex falso quod libet)} \qquad \text{by} \qquad \frac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}\ \text{(reductio ad absurdum)}$$

Basic properties of NJ/NK

- Admissible rules (both in NJ/NK):

$$\frac{\Gamma \vdash A}{\Gamma' \vdash A}\ \Gamma\subseteq\Gamma' \quad \text{(Monotonicity)} \qquad \frac{\Gamma \vdash A}{\Gamma[x := e] \vdash A[x := e]} \quad \text{(Substitutivity)}$$

where $\Gamma \subseteq \Gamma'$ means: for all A, $A \in \Gamma$ implies $A \in \Gamma'$

- From Monotonicity, we deduce (both in NJ/NK):

$$\frac{\Gamma \vdash A}{\sigma\Gamma \vdash A} \quad \text{(Permutation)} \qquad \frac{\Gamma \vdash A}{\Gamma, B \vdash A} \quad \text{(Weakening)} \qquad \frac{\Gamma, B, B \vdash A}{\Gamma, B \vdash A} \quad \text{(Contraction)}$$

- We write $\Gamma \vdash_{\mathrm{NJ}} A$ for: '$\Gamma \vdash A$ is derivable in NJ' (the same for NK)

Proposition (Inclusion NJ $\subseteq$ NK)

If $\Gamma \vdash_{\mathrm{NJ}} A$, then $\Gamma \vdash_{\mathrm{NK}} A$

The axioms of first-order arithmetic

The axioms of first-order arithmetic are the following closed formulas:

- Defining equations of all primitive recursive function symbols:

$$\forall x\,(x+0=x) \qquad \forall x\,(x\times 0=0)$$
$$\forall x\,\forall y\,(x+s(y)=s(x+y)) \qquad \forall x\,\forall y\,(x\times s(y)=x\times y+x)$$
$$\forall x\,(\mathrm{pred}(0)=0) \qquad \forall x\,(x-0=0)$$
$$\forall x\,(\mathrm{pred}(s(x))=x) \qquad \forall x\,\forall y\,(x-s(y))=\mathrm{pred}(x-y)$$

etc.

- Peano axioms:

(P3) $\forall x\,\forall y\,(s(x)=s(y)\Rightarrow x=y)$

(P4) $\forall x\,\neg(s(x)=0)$

(P5) $\forall\vec{z}\,[A(\vec{z},0)\land\forall x\,(A(\vec{z},x)\Rightarrow A(\vec{z},s(x)))\Rightarrow\forall x\,A(\vec{z},x)]$

for all formulas $A(\vec{z},x)$ whose free variables occur among $\vec{z},x$

This set of axioms is written Ax(HA) or Ax(PA)

Heyting Arithmetic (HA)

Definition (Heyting Arithmetic)

Heyting Arithmetic (HA) is the theory based on first-order intuitionistic logic (NJ) and whose set of axioms is Ax(HA). Formally:

$$\mathrm{HA} \vdash A \quad \equiv \quad \Gamma \vdash_{\mathrm{NJ}} A \quad \text{for some } \Gamma \subseteq \mathrm{Ax}(\mathrm{HA})$$

- Replacing NJ by NK, we get Peano Arithmetic (same axioms)
- When building proofs, it is convenient to integrate the axioms of HA in the system of deduction, by replacing the Axiom rule

$$\frac{}{\Gamma \vdash A}\ {}^{A \in \Gamma} \qquad \text{by} \qquad \frac{}{\Gamma \vdash A}\ {}^{A \in \Gamma \cup \mathrm{Ax}(\mathrm{HA})}$$

  The extended deduction system is then written HA

- **Question:** Is HA constructive?

Basic properties

- Given a function symbol f and a closed FO-terms e, we write:
  - $f^{\mathbb{N}}$ ($: \mathbb{N}^k \to \mathbb{N}$) the primitive recursive function associated to f
  - $e^{\mathbb{N}}$ ($\in \mathbb{N}$) the denotation of e in $\mathbb{N}$ (standard model)
- Since the system of axioms of HA provides the defining equations of all primitive recursive functions, we have:

Proposition (Computational completeness)

If $\mathbb{N} \models e_1 = e_2$, then $\mathrm{HA} \vdash e_1 = e_2$

Note: Converse implication amounts to the property of consistency

Corollary (Completeness for Σ^0_1-formulas)

If $\mathbb{N} \models \exists \vec{x}\,(e_1(\vec{x}) = e_2(\vec{x}))$, then $\mathrm{HA} \vdash \exists \vec{x}\,(e_1(\vec{x}) = e_2(\vec{x}))$

Note: Converse implication is the property of 1-consistency

- Gödel's 1st incompleteness theorem says that PA is not Π^0_1-complete

Proving that HA is constructive

We now aim at proving that HA is constructive, in the sense that:

- HA fulfills the disjunction property:

$$\text{If} \quad \mathsf{HA} \vdash A \vee B, \quad \text{then} \quad \mathsf{HA} \vdash A \quad \text{or} \quad \mathsf{HA} \vdash B$$

  (where A, B are closed formulas)

- HA fulfills the witness property:

$$\text{If} \quad \mathsf{HA} \vdash \exists x\, A(x), \quad \text{then} \quad \mathsf{HA} \vdash A(n) \quad \text{for some } n \in \mathbb{N}$$

  (where $A(x)$ only depends on x)

HA is clearly recursive, and consistent from the existence of the standard model

There are essentially two ways to prove this:

(1) As a consequence of a cut elimination theorem

(2) By constructing a realizability model

However, cut elimination is usually deduced from strong normalization, that is most often proved by techniques of realizability

Plan

What is the difference betwen typing and realizability?

Point of view of the hacker:
Many situations where an ill-typed program is correct w.r.t. execution:

```
let my_useless_function n =
  if n * n + 1 = 0 then 42 else true

This expression has type bool but is here used with type int
```

However `my_useless_function` always returns a `bool` when applied to an `int`...

Distinct notions of correctness:

1. Correctness w.r.t. typing
2. Correctness w.r.t. execution $\rightsquigarrow$ Realizability

System T: common parts

Syntax

$$\begin{array}{llcl}
\textbf{Types} & A, B & ::= & \mathsf{nat} \mid A \times B \mid A \to B \\
\textbf{Terms} & t, u & ::= & x \mid \lambda x\,.\,t \mid tu \\
 & & \mid & \langle t_1, t_2 \rangle \mid \pi_1(t) \mid \pi_2(t) \\
 & & \mid & 0 \mid \mathtt{S}(t) \mid \mathtt{rec}(t_0, t_1, u)
\end{array}$$

Reduction rules

$$\begin{array}{rcl}
(\lambda x\,.\,t)u & \succ & t[x := u] \\[1ex]
\pi_1(\langle t_1, t_2 \rangle) & \succ & t_1 \\
\pi_2(\langle t_1, t_2 \rangle) & \succ & t_2 \\[1ex]
\mathtt{rec}(t_0, t_1, 0) & \succ & t_0 \\
\mathtt{rec}(t_0, t_1, \mathtt{S}(u)) & \succ & t_1\ u\ (\mathtt{rec}(t_0, t_1, u))
\end{array}$$

System T: typing (1/2)

$$\frac{}{\Gamma \vdash x : A}\ {\scriptstyle (x:A)\in\Gamma} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x\,.\,t : A \to B} \qquad \frac{\Gamma \vdash t : A \to B \qquad \Gamma \vdash u : A}{\Gamma \vdash tu : B}$$

$$\frac{\Gamma \vdash t_1 : A \qquad \Gamma \vdash t_2 : B}{\Gamma \vdash \langle t_1, t_2\rangle : A \times B} \qquad \frac{\Gamma \vdash t : A \times B}{\Gamma \vdash \pi_1(t) : A} \qquad \frac{\Gamma \vdash t : A \times B}{\Gamma \vdash \pi_2(t) : B}$$

$$\frac{}{\Gamma \vdash 0 : \mathrm{nat}} \qquad \frac{\Gamma \vdash t : \mathrm{nat}}{\Gamma \vdash \mathtt{S}(t) : \mathrm{nat}}$$

$$\frac{\Gamma \vdash t_0 : A \qquad \Gamma \vdash t_1 : \mathrm{nat} \to A \to A \qquad \Gamma \vdash u : A}{\Gamma \vdash \mathtt{rec}(t_0, t_1, u) : A}$$

- Typing deals with *open* terms $\rightsquigarrow$ typing contexts
- Simple justification: typing derivation
- Type checking & inference are decidable (syntax directed)
- But reduction is never mentioned...

  ... which guaranty do we have w.r.t. computation?

System T: typing (2/2)

3 results ensure the correctness of the system w.r.t. computation:

(1) Subject reduction

If $\Gamma \vdash t : A$ and $t \succ t'$, then $\Gamma \vdash t' : A$

(2) Closed normal forms of type nat

If $\vdash t : \mathrm{nat}$, where t is in normal form, then $t = \mathrm{S}^n(0)$ (for some n)

(3) Normalization

If $\Gamma \vdash t : A$, then t is (strongly) normalizing

$(1) + (2) + (3) \quad \Rightarrow$ Every closed term $t : \mathrm{nat}$ reduces to a natural

Remarks: Proofs of (1) and (2) are purely combinatorial. Proof of (3) is in general not combinatorial, and usually relies on realizability techniques

System T: realizability

Binary relation $t \Vdash A$ (t = closed term)

Definition of the realizability relation

1. $t \Vdash \mathrm{nat}$ iff $t \succ^* \mathrm{S}^n(0)$ for some $n \in \mathbb{N}$
2. $t \Vdash A \times B$ iff $t \succ^* \langle t_1, t_2 \rangle$ for some $t_1 \Vdash A$ y $t_2 \Vdash B$
3. $t \Vdash A \to B$ iff $u \Vdash A$ implies $tu \Vdash B$ (for all u)

- Closed terms $\rightsquigarrow$ no context
- Purely computational definition: syntax = black box
- No correctness to prove; hard-wired in the definition!

  (Set of realizers of A is closed under anti-reduction)
- No elementary justification (as a derivation)

  $\rightsquigarrow$ requires an external justification: proof
- Relation $t \Vdash A$ is undecidable, not even recursively enumerable

System T: from typing to realizability

Theorem (Adequacy)

If $x_1 : A_1, \ldots, x_n : A_n \vdash t : B$, then for all $u_1, \ldots, u_n$:

$u_1 \Vdash A_1, \ \ldots, \ u_n \Vdash A_n$ imply $t[x_1 := u_1; \ldots; x_n := u_n] \Vdash B$

Proof: straightforward induction on the derivation.
The cases of λ and $\langle \_, \_ \rangle$ rely on the property of closure under anti-reduction.

Particular case (empty context): $\vdash t : A$ implies $t \Vdash A$

- Typing + adequacy $\leadsto$
  every closed term t : nat reduces to a natural

  (Without using (1) + (2) + (3). Actually, (3) is proved by realizability.)

- **Beware!** $t \Vdash A$ does not imply $t : A$
  There are much more realizers than well-typed terms

Example of an ill-typed realizer

- In system T, we easily implement a term $\texttt{is\_prime} : \mathsf{nat} \to \mathsf{nat}$ such that

$$\texttt{is\_prime}\ (\mathtt{S}^n(0)) \succ^* \begin{cases} \mathtt{S}(0) & \text{if } n \text{ is prime} \\ 0 & \text{otherwise} \end{cases}$$

- From this, we let:

$$\begin{aligned} Y &:\equiv (\lambda w\,.\,\lambda f\,.\,f\,(w\,w\,f))\,(\lambda w\,.\,\lambda f\,.\,f\,(w\,w\,f)) \\ \texttt{if}(u, t_1, t_0) &:\equiv \texttt{rec}(t_0,\ \lambda\_\,.\,\lambda\_\,.\,t_1,\ u) \\ \texttt{next\_prime} &:\equiv Y\ (\lambda f\,.\,\lambda x\,.\,\texttt{if}(\texttt{is\_prime}(\texttt{x}),\ x,\ f\ (\mathtt{S}(x)))) \end{aligned}$$

- Clearly: $\texttt{next\_prime} \Vdash \mathsf{nat} \to \mathsf{nat}$

  $\nvdash\ \texttt{next\_prime} : \mathsf{nat} \to \mathsf{nat}$

(Since next_prime contains non-normalizing subterm Y)

Intermezzo: syntax vs semantics

All of this is reminiscent from a well-known phenomenon in logic:

- If $\mathscr{M}$ is a Tarski model of a theory $\mathscr{T}$:

$$\mathscr{T} \vdash \phi \qquad \Rightarrow \qquad \mathscr{M} \models \phi \qquad\qquad (\text{but } \not\Leftarrow)$$

- If $\mathscr{M}$ is a realizability model of a type system $\mathscr{T}$:

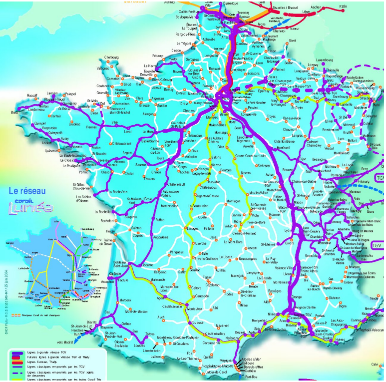
$$\vdash_{\mathscr{T}} t : A \qquad \Rightarrow \qquad t \Vdash_{\mathscr{M}} A \qquad\qquad (\text{but } \not\Leftarrow)$$

Semantics always captures more judgments than syntax...
... but decidability/recursivity is irremediably lost

In logic:

Realizability = model theory based on the operational semantics of the underlying "proofs" (intuitionistic or classical)

Typing and realizability: a metaphore



Plan

Background

- 1908. Brouwer: *The untrustworthiness of the principles of logic*
  (Principles of intuitionism)
- 1936. Church: *An unsolvable problem of elementary number theory*
  (Application of the λ-calculus to the *Entscheidungsproblem*)
- 1936. Turing: *On computable numbers, with an application to the Entscheidungsproblem*
  (Alternative solution to the *Entscheidungsproblem*, using Turing machines)
- 1936. Kleene: *λ-definability and recursiveness*
  (Definition of partial recursive functions)
- 1945. Kleene: *On the interpretation of intuitionistic number theory*
  (Introduction of realizability, as a semantics for HA)

A survey on (intuitionistic) realizability:
Jaap van Oosten: *Realizability: A Historical Essay.*
Mathematical Structures in Computer Science 12(3): 239–263. 2002

The Brouwer-Heyting-Kolmogorov (BHK) semantics

- **Philosophical input:** the meaning of a proposition A is the set $[\![A]\!]$ of "evidences" that A holds:

$$[\![A \wedge B]\!] = [\![A]\!] \times [\![B]\!] \qquad \text{(Cartesian product)}$$

$$[\![A \vee B]\!] = [\![A]\!] + [\![B]\!] \qquad \text{(disjoint union)}$$

$$[\![\bot]\!] = \varnothing \qquad \text{(empty set of evidences)}$$

$$[\![A \Rightarrow B]\!] = [\![A]\!] \to [\![B]\!] \qquad \text{('computable' functions)}$$

$$[\![\forall x\, A(x)]\!] = \prod_{x \in D} [\![A(x)]\!] \qquad \text{(dependent product)}$$

$$[\![\exists x\, A(x)]\!] = \sum_{x \in D} [\![A(x)]\!] \qquad \text{(dependent sum)}$$

(where D is the semantic domain of quantifications)

- The BHK philosophical interpretation of propositions can be given a mathematical contents: the theory of realizability

Kleene realizability

1945. Kleene: *On the interpretation of intuitionistic number theory*

- Realizability in Heyting Arithmetic (HA)
- Definition of the realizability relation $n \Vdash A$
  - n = Gödel code of a partial recursive function
  - A = closed formula of HA
- **Theorem:** Every provable formula of HA is realized
  (But some unprovable formulas are realized too...)
- Application to the disjunction & existence properties

Remarks:

- Codes for partial recursive functions can be replaced by the elements of any partial combinatory algebra (see later)
- Here, we shall take the closed terms of (untyped) system T

The language of realizers

Terms of system T (= λ-calculus + primitive pairs & integers)

$$
\begin{array}{llll}
\lambda\textbf{-terms} & t, u & ::= & x \mid \lambda x\,.\,t \mid tu \\
& & \mid & \langle t_1, t_2\rangle \mid \pi_1(t) \mid \pi_2(t) \\
& & \mid & 0 \mid \mathtt{S}(t) \mid \mathtt{rec}(t_0, t_1, u)
\end{array}
$$

Syntactic worship: Free & bound variables. Renaming. Work up to α-conversion. Set of free variables: $FV(t)$. Capture-avoiding substitution: $t[x := u]$

- **Notation:** $\bar{n} \;:=\; \mathtt{S}^n\,0 \quad (n \in \mathbb{N})$

Reduction rules

$$(\lambda x\,.\,t)\,u \;\succ\; t[x := u]$$

$$
\begin{array}{lllllll}
\pi_1(\langle t_1, t_2\rangle) & \succ & t_1 & \qquad & \mathtt{rec}(t_0,\, t_1,\, 0) & \succ & t_0 \\
\pi_2(\langle t_1, t_2\rangle) & \succ & t_2 & \qquad & \mathtt{rec}(t_0,\, t_1\,\mathtt{S}(u)) & \succ & t_1\;u\;(\mathtt{rec}(t_0, t_1, u))
\end{array}
$$

- Grand reduction written $t \succ^* u$ (reflexive, transitive, context-closed)

Definition of the relation $t \Vdash A$

- **Recall:** For each closed FO-term e, we write $e^{\mathbb{N}}$ its denotation in $\mathbb{N}$

Definition of the realizability relation $t \Vdash A$ (t, A closed)

$$
\begin{array}{lcl}
t \Vdash \bot & \equiv & \bot \\
t \Vdash \top & \equiv & t \succ^* 0 \\
t \Vdash e_1 = e_2 & \equiv & e_1^{\mathbb{N}} = e_2^{\mathbb{N}} \ \wedge \ t \succ^* 0 \\
t \Vdash A \wedge B & \equiv & \exists t_1 \, \exists t_2 \, (t \succ^* \langle t_1, t_2 \rangle \ \wedge \ t_1 \Vdash A \ \wedge \ t_2 \Vdash B) \\
t \Vdash A \vee B & \equiv & \exists u \, ((t \succ^* \langle \bar{0}, u \rangle \ \wedge \ u \Vdash A) \ \vee \ (t \succ^* \langle \bar{1}, u \rangle \ \wedge \ u \Vdash B)) \\
t \Vdash A \Rightarrow B & \equiv & \forall u \, (u \Vdash A \ \Rightarrow \ tu \Vdash B) \\
t \Vdash \forall x \, A(x) & \equiv & \forall n \, (t \, \bar{n} \Vdash A(n)) \\
t \Vdash \exists x \, A(x) & \equiv & \exists n \, \exists u \, (t \succ^* \langle \bar{n}, u \rangle \ \wedge \ u \Vdash A(n))
\end{array}
$$

Lemma (Closure under anti-reduction)

If $t \succ^* t'$ and $t' \Vdash A$, then $t \Vdash A$

We now want to prove the

Theorem (Soundness)

If HA $\vdash A$, then $t \Vdash A$ for some closed λ-term t

Outline of the proof:

- **Step 1:** Translating FO-terms into λ-terms
- **Step 2:** Translating derivations of LJ into λ-terms
- **Step 3:** Adequacy lemma
- **Step 4:** Realizing the axioms of HA
- **Final step:** Putting it all together

Step 1: Translating FO-terms into λ-terms

Proposition (Compiling primitive recursive functions in system T)

Each (prim. rec.) function symbol f is computed by a closed λ-term f^*:

$$\text{If} \quad f^{\mathbb{N}}(n_1, \ldots, n_k) = m, \quad \text{then} \quad f^*\ \bar{n}_1 \cdots \bar{n}_k \ \succ^* \ \bar{m}$$

Proof. Standard exercise of compilation. Examples:

$$\begin{array}{rcl} 0^* & := & 0 \\ s^* & := & \mathtt{S} \\ \mathrm{pred}^* & := & \lambda x\,.\,\mathtt{rec}(0,\ \lambda z, \_\,.\,z,\ x) \end{array} \qquad \begin{array}{rcl} (+)^* & := & \lambda x, y\,.\,\mathtt{rec}(x,\ \lambda \_, z\,.\,\mathtt{S}\,z,\ y) \\ (\times)^* & := & \lambda x, y\,.\,\mathtt{rec}(0,\ \lambda \_, z\,.\,(+)^*\ z\ x,\ y) \\ (-)^* & := & \lambda x, y\,.\,\mathtt{rec}(x,\ \lambda \_, z\,.\,\mathrm{pred}^*\ z,\ y) \end{array}$$

- Each FO-term e with free variables $x_1, \ldots, x_k$ is translated into a closed λ-term e^* with the same free variables, letting:

$$x^* \ := \ x \qquad \text{and} \qquad \big(f(e_1, \ldots, e_k)\big)^* \ := \ f^*\ e_1^* \cdots e_k^*$$

Fact: If e is closed, then $e^* \ \succ^* \ \bar{n}$, where $n = e^{\mathbb{N}}$

Step 2: Translating derivations into λ-terms (1/3)

- Every derivation $d : (A_1, \ldots, A_n \vdash B)$ is translated into a λ-term d^* with free variables $x_1, \ldots, x_k, z_{A_1}, \ldots, z_{A_n}$, where:
  - $x_1, \ldots, x_k$ are the free variables of $A_1, \ldots, A_n, B$
  - $z_{A_1}, \ldots, z_{A_n}$ are proof variables associated to hypotheses $A_1, \ldots, A_n$
- The construction of d^* follows the Curry-Howard correspondence:

$$\left(\frac{}{A_1, \ldots, A_n \vdash A_i}\right)^* := z_{A_i} \qquad \left(\frac{}{\Gamma \vdash \top}\right)^* := 0 \qquad \left(\frac{\begin{array}{c}\vdots\ d\\ \Gamma \vdash \bot\end{array}}{\Gamma \vdash A}\right)^* := \text{any\_term}$$

$$\left(\frac{\begin{array}{c}\vdots\ d\\ \Gamma, A \vdash B\end{array}}{\Gamma \vdash A \Rightarrow B}\right)^* := \lambda z_A \,.\, d^* \qquad \left(\frac{\begin{array}{c}\vdots\ d_1\\ \Gamma \vdash A \Rightarrow B\end{array} \quad \begin{array}{c}\vdots\ d_2\\ \Gamma \vdash A\end{array}}{\Gamma \vdash B}\right)^* := d_1^* d_2^*$$

Step 2: Translating derivations into λ-terms (2/3)

$$\left(\dfrac{\begin{array}{c}\vdots\; d_1 \\ \Gamma \vdash A\end{array} \quad \begin{array}{c}\vdots\; d_2 \\ \Gamma \vdash B\end{array}}{\Gamma \vdash A \land B}\right)^* \;:=\; \langle d_1^*, d_2^* \rangle$$

$$\left(\dfrac{\begin{array}{c}\vdots\; d \\ \Gamma \vdash A \land B\end{array}}{\Gamma \vdash A}\right)^* \;:=\; \pi_1(d^*) \qquad \left(\dfrac{\begin{array}{c}\vdots\; d \\ \Gamma \vdash A \land B\end{array}}{\Gamma \vdash B}\right)^* \;:=\; \pi_2(d^*)$$

$$\left(\dfrac{\begin{array}{c}\vdots\; d \\ \Gamma \vdash A\end{array}}{\Gamma \vdash A \lor B}\right)^* \;:=\; \langle \bar{0}, d^* \rangle \qquad \left(\dfrac{\begin{array}{c}\vdots\; d \\ \Gamma \vdash B\end{array}}{\Gamma \vdash A \lor B}\right)^* \;:=\; \langle \bar{1}, d^* \rangle$$

$$\left(\dfrac{\begin{array}{c}\vdots\; d \\ \Gamma \vdash A \lor B\end{array} \quad \begin{array}{c}\vdots\; d_1 \\ \Gamma, A \vdash C\end{array} \quad \begin{array}{c}\vdots\; d_2 \\ \Gamma, B \vdash C\end{array}}{\Gamma \vdash C}\right)^* \;:=\; \mathsf{match}\big(d^*,\ \lambda z_A \,.\, d_1^*,\ \lambda z_B \,.\, d_2^*\big)$$

writing $\quad \mathsf{match}(u, t_1, t_2) \;:=\; \mathtt{rec}\big(t_1\,(\pi_2(u)),\ \lambda\_,\_\,.\, t_2\,(\pi_2(u)),\ \pi_1(u)\big)$

Step 2: Translating derivations into λ-terms (3/3)

$$\left(\dfrac{\begin{array}{c}\vdots\ d\\ \Gamma\vdash A\end{array}}{\Gamma\vdash\forall x\,A}\right)^{*} := \lambda x\,.\,d^{*} \qquad \left(\dfrac{\begin{array}{c}\vdots\ d\\ \Gamma\vdash\forall x\,A\end{array}}{\Gamma\vdash A[x:=e]}\right)^{*} := d^{*}e^{*}$$

$$\left(\dfrac{\begin{array}{c}\vdots\ d\\ \Gamma\vdash A[x:=e]\end{array}}{\Gamma\vdash\exists x\,A}\right)^{*} := \langle e^{*},d^{*}\rangle \qquad \left(\dfrac{\begin{array}{cc}\vdots\ d_1 & \vdots\ d_2\\ \Gamma\vdash\exists x\,A & \Gamma,A\vdash B\end{array}}{\Gamma\vdash B}\right)^{*} := \text{let }\langle x,z\rangle=d_1^{*}\text{ in }d_2^{*}$$

$$\left(\overline{\Gamma\vdash e=e}\right)^{*} := 0 \qquad \left(\dfrac{\begin{array}{cc}\vdots\ d_1 & \vdots\ d_2\\ \Gamma\vdash e_1=e_2 & \Gamma\vdash A[x=e_1]\end{array}}{\Gamma\vdash A[x:=e_2]}\right)^{*} := d_2^{*}$$

writing $\quad \text{let }\langle x,z\rangle=t\text{ in }u \;:=\; (\lambda y\,.\,(\lambda x,z\,.\,u)\ \pi_1(y)\ \pi_2(y))\ t$

Step 3: Adequacy lemma

Recall that in the definition of d^*, we assumed that each first-order variable x is also a λ-variable. (Remaining λ-variables z are used as proof variables.)

Definition (Valuation)

A valuation is a function $\rho : \mathrm{FOVar} \to \mathbb{N}$. A valuation ρ may be applied:

- to a formula A; notation: $A[\rho]$ (result is a closed formula)
- to a λ-term t; notation: $t[\rho]$ (result is a possibly open λ-term)

Lemma (Adequacy)

Let $d : (A_1, \ldots, A_n \vdash B)$ be a derivation in NJ. Then:

- for all valuations ρ,
- for all realizers $t_1 \Vdash A_1[\rho], \ldots, t_n \Vdash A_n[\rho]$,

we have: $$d^*[\rho][z_1 := t_1, \ldots, z_n := t_n] \Vdash B[\rho]$$

Proof: By induction on d, using that $\{t : t \Vdash B\}$ is closed under anti-evaluation

Step 4: Realizing the axioms of HA

Lemma (Realizing true Π^0_1-formulas)

Let $e_1(\vec{x})$, $e_2(\vec{x})$ be FO-terms depending on free variables $\vec{x}$.

If $\mathbb{N} \models \forall \vec{x}\,(e_1(\vec{x}) = e_2(\vec{x}))$, then $\lambda \vec{x}\,.\,\bar{0} \Vdash \forall \vec{x}\,(e_1(\vec{x}) = e_2(\vec{x}))$

- Since all defining equations of function symbols are Π^0_1:

Corollary

All defining equations of function symbols are realized

Lemma (Realizing Peano axioms)

$$\begin{array}{rcl} \lambda xyz\,.\,z & \Vdash & \forall x\,\forall y\,(s(x) = s(y) \Rightarrow x = y) \\ \text{any\_term} & \Vdash & \forall x\,(s(x) \neq 0) \\ \lambda \vec{y}\,.\,\mathtt{rec} & \Vdash & \forall \vec{y}\,[A(\vec{y},0) \Rightarrow \forall x\,(A(\vec{y},x) \Rightarrow A(\vec{y},s(x))) \Rightarrow \forall x\,A(\vec{y},x)] \end{array}$$

writing $\mathtt{rec} \;:\equiv\; \lambda z_0, z_1, x\,.\,\mathtt{rec}(z_0, z_1, x)$

Final step: Putting it all together

Theorem (Soundness)

If HA $\vdash A$, then $t \Vdash A$ for some closed λ-term t

Proof. Assume HA $\vdash A$, so that there are axioms $A_1, \ldots, A_n$ and a derivation $d : (A_1, \ldots, A_n \vdash A)$ in LJ. Take realizers $t_1, \ldots, t_n$ of $A_1, \ldots, A_n$. By adequacy, we have $d^*[z_1 := t_1, \ldots, z_n := t_n] \Vdash A$.

Corollary (Consistency)

HA is consistent: HA $\not\vdash \bot$

Proof. If HA $\vdash \bot$, then the formula $\bot$ is realized, which is impossible by definition

- **Remark.** Since HA $\subseteq$ PA and PA is consistent (from the existence of the standard model), we already knew that HA is consistent

Σ^0_1-soundness and completeness

Proposition (Σ^0_1-soundness/completeness)

For every closed Σ^0_1-formula, the following are equivalent:

(1) $\mathrm{HA} \vdash \exists\vec{x}\,(e_1(\vec{x}) = e_2(\vec{x}))$ (formula is provable)

(2) $t \Vdash \exists\vec{x}\,(e_1(\vec{x}) = e_2(\vec{x}))$ for some t (formula is realized)

(3) $\mathbb{N} \models \exists\vec{x}\,(e_1(\vec{x}) = e_2(\vec{x}))$ (formula is true)

Proof. (1) $\Rightarrow$ (2) by soundness
(2) $\Rightarrow$ (3) by definition of $t \Vdash \exists\vec{x}\,(e_1(\vec{x}) = e_2(\vec{x}))$
(3) $\Rightarrow$ (1) by Σ^0_1-completeness

Corollary (Existence property for Σ^0_1-formulas)

If $\mathrm{HA} \vdash \exists\vec{x}\,(e_1(\vec{x}) = e_2(\vec{x}))$, then $\mathrm{HA} \vdash e_1(\vec{n}) = e_2(\vec{n})$ for some $\vec{n} \in \mathbb{N}$

Proof. Use (1) $\Rightarrow$ (3), and conclude by computational completeness

The halting problem

- Let h be the binary function symbol associated to the primitive recursive function $h^{\mathbb{N}} : \mathbb{N}^2 \to \mathbb{N}$ defined by

$$h^{\mathbb{N}}(n,k) \;=\; \begin{cases} 1 & \text{if Turing machine } n \text{ stops after } k \text{ evaluation steps} \\ 0 & \text{otherwise} \end{cases}$$

- Write $H(x) \;:=\; \exists y\,(h(x,y) = 1)$ (halting predicate)

Proposition

The formula $\forall x\,(H(x) \vee \neg H(x))$ is **not** realized

Proof. Let $t \Vdash \forall x\,(H(x) \vee \neg H(x))$, and put $u := \lambda x\,.\,\mathtt{fst}\,(t\,x)$. We check that:

- For all $n \in \mathbb{N}$, either $u\,\bar{n} \succ^* \bar{0}$ or $u\,\bar{n} \succ^* \bar{1}$
- If $u\,\bar{n} \succ^* \bar{0}$, then $H(n)$ is realized, so that Turing machine n halts
- If $u\,\bar{n} \succ^* \bar{1}$, then $H(n)$ is not realized so that Turing machine n loops

Therefore, the program u solves the halting problem, which is impossible

EM is not derivable in HA

- By soundness we get: HA $\nvdash$ $\forall x\,(H(x) \vee \neg H(x))$. Hence:

Theorem (Unprovability of EM)

The law of excluded middle (EM) is not provable in HA

- **Remark:** We actually showed that the open instance $H(x) \vee \neg H(x)$ of EM is not provable in HA. On the other hand we can prove (classically) that each closed instance of EM is realizable:

Proposition (Realizing closed instances of EM)

For each **closed** formula A, the formula $A \vee \neg A$ is realized

Proof. Using meta-theoretic EM (in the model), we distinguish two cases:

- Either A is realized by some term t. Then $\langle \bar{0}, t\rangle \Vdash A \vee \neg A$
- Either A is not realized. Then $t \Vdash \neg A$ (t any), hence $\langle \bar{1}, t\rangle \Vdash A \vee \neg A$

- But this proof is not accepted by intuitionists (uses meta-theoretic EM)

Unprovable, but realizable (1/3)

- We have already seen that the Halting Problem

(HP) $\forall x\,(H(x) \vee \neg H(x))$

is not realized. Therefore:

Proposition

any_term $\Vdash \neg$HP, but: HA $\nvdash \neg$HP (since: PA $\nvdash \neg$HP)

Proof. Since HP is not realized, its negation is realized by any term. On the other hand we have PA $\nvdash \neg$HP (since PA $\vdash$ HP), so that HA $\nvdash \neg$HP

- **Morality:**
  - PA takes position for the excluded middle
  - HA actually takes no position (for or against) the excluded middle. In practice, it is 100% compatible with classical logic
  - Kleene realizability takes position against excluded middle. Many realized formulas (such as $\neg$HP) are classically false

Unprovable, but realizable (2/3)

- Recall that all true Π^0_1-formulas are realized:

$$\text{If} \quad \mathbb{N} \models \forall \vec{x}\,(e_1(\vec{x}) = e_2(\vec{x})), \quad \text{then} \quad \lambda \vec{x}\,.\,\bar{0} \Vdash \forall \vec{x}\,(e_1(\vec{x}) = e_2(\vec{x}))$$

- But Gödel undecidable formula G is a true Π^0_1-formula. Therefore:

Proposition

$\lambda z\,.\,\bar{0} \Vdash G$, but: $\mathrm{HA} \nvdash G$ (since: $\mathrm{PA} \nvdash G$)

Remarks:

- Like $\neg$HP, the formula G is realized but not provable
- Unlike $\neg$HP, the formula G is classically true

Unprovable, but realizable (3/3)

- Markov Principle (MP) is the following scheme of axioms:

$$\forall x\,(A(x) \vee \neg A(x)) \Rightarrow \neg\neg\exists x\,A(x) \Rightarrow \exists x\,A(x)$$

- Obviously: $\mathsf{PA} \vdash \mathsf{MP}$

Proposition (MP is realized)

$$t_{\mathsf{MP}} \Vdash \forall x\,(A(x) \vee \neg A(x)) \Rightarrow \neg\neg\exists x\,A(x) \Rightarrow \exists x\,A(x)$$

where

$$t_{\mathsf{MP}} := \lambda z\_\,.\,\mathbf{Y}\,(\lambda rx\,.\,\text{if } \mathtt{fst}\,(z\,x) = 0 \text{ then } \langle x, \mathtt{snd}\,(z\,x)\rangle \text{ else } r\,(\mathtt{S}\,x))$$

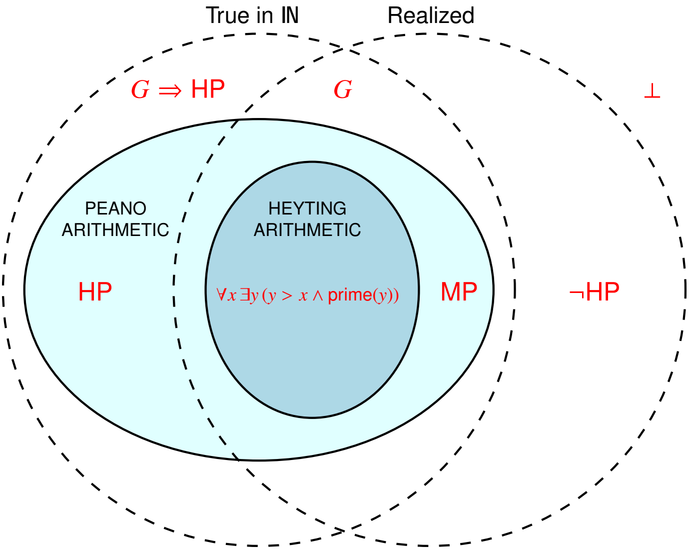
$$\mathbf{Y} := \lambda f\,.\,(\lambda x\,.\,f\,(x\,x))\,(\lambda x\,.\,f\,(x\,x))$$

- Using modified realizability, one can show: $\mathsf{HA} \nvdash \mathsf{MP}$ [Kreisel]

- We have the strict inclusions:

$$\mathsf{HA} \subset \mathsf{HA} + \mathsf{MP} \subset \mathsf{PA}$$

To sum up


True in ℕ
Realized
G ⇒ HP
G
⊥
PEANO ARITHMETIC
HEYTING ARITHMETIC
HP
∀x ∃y (y > x ∧ prime(y))
MP
¬HP

Towards the disjunction and existence properties

Proposition (Semantic disjunction & existence properties)

1. If $\mathrm{HA} \vdash A \vee B$, then A is realized or B is realized
2. If $\mathrm{HA} \vdash \exists x\, A(x)$, then $A(n)$ is realized for some $n \in \mathbb{N}$

Proof. From main Theorem & definition of realizability:

1. If $\mathrm{HA} \vdash A \vee B$, then $t \Vdash A \vee B$ for some t, so that: either $t \succ^* \langle \bar{0}, u \rangle$ for some $u \Vdash A$, or $t \succ^* \langle \bar{1}, u \rangle$ for some $u \Vdash B$
2. If $\mathrm{HA} \vdash \exists x\, A(x)$, then $t \Vdash \exists x\, A(x)$ for some t, so that: $t \succ^* \langle \bar{n}, u \rangle$ for some $n \in \mathbb{N}$ and $u \Vdash A(n)$

- These weak forms of the disjunction & existence properties are now widely accepted as criteria of constructivity
- To prove the strong forms of the disjunction and existence properties (criteria (3) and (4) = (5)), we need to introduce glued realizability

Glued realizability (1/3)

- Let $\mathcal{P}$ be a set of closed formulas such that:
  - $\mathcal{P}$ contains all theorems of HA
  - $\mathcal{P}$ is closed under modus ponens: $(A \Rightarrow B) \in \mathcal{P},\ A \in \mathcal{P} \ \Rightarrow\ B \in \mathcal{P}$

Definition of the relation $t \Vdash_{\mathcal{P}} A$ (t, A closed)

$$
\begin{array}{lcl}
t \Vdash_{\mathcal{P}} \bot & \equiv & \bot \\
t \Vdash_{\mathcal{P}} \top & \equiv & t \succ^* 0 \\
t \Vdash_{\mathcal{P}} e_1 = e_2 & \equiv & e_1^{\mathbb{N}} = e_2^{\mathbb{N}} \ \wedge\ t \succ^* 0 \\
t \Vdash_{\mathcal{P}} A \wedge B & \equiv & \exists t_1\, \exists t_2\, (t \succ^* \langle t_1, t_2 \rangle \ \wedge\ t_1 \Vdash_{\mathcal{P}} A \ \wedge\ t_2 \Vdash_{\mathcal{P}} B) \\
t \Vdash_{\mathcal{P}} A \vee B & \equiv & \exists u\, ((t \succ^* \langle \bar{0}, u \rangle \ \wedge\ u \Vdash_{\mathcal{P}} A) \ \vee\ (t \succ^* \langle \bar{1}, u \rangle \ \wedge\ u \Vdash_{\mathcal{P}} B)) \\
t \Vdash_{\mathcal{P}} A \Rightarrow B & \equiv & \forall u\, (u \Vdash_{\mathcal{P}} A \ \Rightarrow\ tu \Vdash_{\mathcal{P}} B) \quad \wedge \quad (A \Rightarrow B) \in \mathcal{P} \\
t \Vdash_{\mathcal{P}} \forall x\, A(x) & \equiv & \forall n\, (t\, \bar{n} \Vdash_{\mathcal{P}} A(n)) \quad \wedge \quad (\forall x\, A(x)) \in \mathcal{P} \\
t \Vdash_{\mathcal{P}} \exists x\, A(x) & \equiv & \exists n\, \exists u\, (t \succ^* \langle \bar{n}, u \rangle \ \wedge\ u \Vdash_{\mathcal{P}} A(n))
\end{array}
$$

- Plain realizability $=$ case where $\mathcal{P}$ contains all closed formulas

Glued realizability (2/3)

Theorem [Kleene'45]

1. If $t \Vdash_{\mathcal{P}} A$, then $A \in \mathcal{P}$ ($\mathcal{P}$-realizability is "bounded" by $\mathcal{P}$)
2. If $\mathrm{HA} \vdash A$, then $t \Vdash_{\mathcal{P}} A$ for some λ-term t (Adequacy)

Proof.

1. By a straightforward induction on A
2. Same proof as for plain realizability. Extracted program t is the same as before (definitions of $f \mapsto f^*$, $e \mapsto e^*$, $d \mapsto d^*$ unchanged). Only change appears in the statement & proof of Adequacy (step 3), that uses $\Vdash_{\mathcal{P}}$ rather than $\Vdash$.

- **To sum up:** For each set of closed formulas $\mathcal{P}$ that contains all theorems of HA and that is closed under modus ponens:

$$\text{provable in HA} \quad \subseteq \quad \mathcal{P}\text{-realized} \quad \subseteq \quad \mathcal{P}$$

Glued realizability (3/3)

- **Particular case:** $\mathcal{P} = \text{HA}$: (= set of theorems of HA)

Proposition

$\text{HA} \vdash A$ iff $t \Vdash_{\text{HA}} A$ for some closed λ-term t

- From this we deduce:

Corollary (Disjunction/existence properties)

1. If $\text{HA} \vdash A \vee B$, then $\text{HA} \vdash A$ or $\text{HA} \vdash B$
2. If $\text{HA} \vdash \exists x\, A(x)$, then $\text{HA} \vdash A(n)$ for some $n \in \mathbb{N}$

Proof. Same proof as before, using the fact that $\text{HA} \vdash A$ iff A is HA-realized

- **Conclusion:** We proved that HA is constructive, *champagne!*

Program extraction

Proposition (Provably total functions are recursive)

If $\text{HA} \vdash \forall\vec{x}\,\exists y\, A(\vec{x}, y)$ (i.e. the relation $A(\vec{x}, y)$ is provably total in HA), then there exists a total recursive function $\phi : \mathbb{N}^k \to \mathbb{N}$ such that:

$$\text{HA} \vdash A(\vec{n}, \phi(\vec{n})) \qquad \text{for all } \vec{n} = (n_1, \ldots, n_k) \in \mathbb{N}^k$$

Proof. Let d be a derivation of A in HA, and d^* the corresponding closed λ-term (constructed in Steps 1, 2, 4). We take $\phi := \lambda\vec{x}\,.\,\pi_1(d^*\,\vec{x})$

- **Note:** The relation $A(\vec{x}, y)$ may not be functional. In this case, the extracted program $\phi := \lambda\vec{x}\,.\,\pi_1(d^*\,\vec{x})$ associated to the derivation d chooses one output $\phi(\vec{n})$ for each input $\vec{n} \in \mathbb{N}^k$
- **Optimizing extracted program ϕ:** Using modified realizability [Kreisel], we can ignore all sub-proofs corresponding to Harrop formulas:

Harrop formulas

$$H \;::=\; e_1 = e_2 \;\mid\; \top \;\mid\; \bot \;\mid\; H_1 \wedge H_2 \;\mid\; A \Rightarrow H \;\mid\; \forall x\, H$$

Plan

Kleene's original presentation (1/2)

- Kleene did not consider closed λ-terms as realizers, but natural numbers, used as Gödel codes for partial recursive functions
- Definition of realizability parameterized by:
  - A recursive injection $\langle\cdot,\cdot\rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ (pairing)
  - An enumeration $(\phi_n)_{n\in\mathbb{N}}$ of all partial recursive functions of arity 1
- Kleene application: $n \cdot p \ := \ \phi_n(p)$ (partial operation)
- **Realizability relation:** $n \Vdash A$ ($n \in \mathbb{N}$, A closed formula)

Theorem

If $\mathrm{HA} \vdash A$, then $n \Vdash A$ for some $n \in \mathbb{N}$

- As before, we can also realize many unprovable formulas, such as the negation of the Halting Problem ($\neg$HP), Gödel undecidable formula G and Markov Principle (MP), as well as Church's Thesis (CT) (cf later)

Kleene's original presentation (2/2)

Definition of the realizability relation $n \Vdash A$ ($n \in \mathbb{N}$, A closed)

$$
\begin{array}{lcl}
n \Vdash \bot & \equiv & \bot \\
n \Vdash \top & \equiv & n = 0 \\
n \Vdash e_1 = e_2 & \equiv & e_1^{\mathbb{N}} = e_2^{\mathbb{N}} \ \wedge \ n = 0 \\
n \Vdash A \wedge B & \equiv & \exists n_1 \ \exists n_2 \ (n = \langle n_1, n_2 \rangle \ \wedge \ n_1 \Vdash A \ \wedge \ n_2 \Vdash B) \\
n \Vdash A \vee B & \equiv & \exists m \ ((n = \langle 0, m \rangle \ \wedge \ m \Vdash A) \ \vee \ (n = \langle 1, m \rangle \ \wedge \ m \Vdash B)) \\
n \Vdash A \Rightarrow B & \equiv & \forall p \ (p \Vdash A \ \Rightarrow \ n \cdot p \Vdash B) \\
n \Vdash \forall x \, A(x) & \equiv & \forall p \ (n \cdot p \Vdash A(p)) \\
n \Vdash \exists x \, A(x) & \equiv & \exists p \ \exists m \ (n = \langle p, m \rangle \ \wedge \ m \Vdash A(p))
\end{array}
$$

- Proof of Main Theorem is essentially the same as before. But:
  - We need to work with Hilbert's system for LJ (rather than with NJ)
  - Gödel codes induce a lot of code obfuscation...
- As before, we can define glued realizability, prove the disjunction & existence properties, extract program from proofs, etc.

Church's Thesis (CT)

- Let h' be the ternary function symbol associated to the primitive recursive function $h'^{\mathbb{N}} : \mathbb{N}^3 \to \mathbb{N}$ defined by

$$h'^{\mathbb{N}}(n,p,k) = \begin{cases} s(r) & \text{if Turing machine } n \text{ applied to } p \text{ stops after} \\ & k \text{ evaluation steps and returns } r \\ 0 & \text{otherwise} \end{cases}$$

and put: $\quad x \cdot y = z \quad := \quad \exists k\,(h'(x,y,k) = s(z))$

- Church's Thesis (CT) internalizes in the language of HA the fact that every provably total function is recursive:

$$\text{(CT)} \qquad \forall x\,\exists y\,A(x,y) \;\Rightarrow\; \exists n\,\forall x\,\exists y\,(n \cdot x = y \wedge A(x,y))$$

- Clearly: $\quad \mathrm{PA} \vdash \neg\mathrm{CT} \qquad$ (take $A(x,y) \;:=\; (H(x) \wedge y = 1) \vee (\neg H(x) \wedge y = 0)$)

Proposition

CT is realized by some $n \in \mathbb{N}$ (although $\mathrm{HA} \nvdash \mathrm{CT}$)

Towards partial combinatory algebras

Idea: To define a language of realizers, we need a set $\mathcal{A}$ whose elements behave as partial functions on $\mathcal{A}$, and that is 'closed under λ-abstraction'

Definition (Partial applicative structure – PAS)

A partial applicative structure (PAS) is a set $\mathcal{A}$ equipped with a partial function $(\cdot) \;:\; \mathcal{A} \times \mathcal{A} \rightharpoonup \mathcal{A}$, called application

Notation: $abc = (a \cdot b) \cdot c$, etc. (application is left-associative)

- **Intuition:** Each element a of a partial applicative structure $\mathcal{A}$ represents a partial function on $\mathcal{A}$: $(b \mapsto ab) \;:\; \mathcal{A} \rightharpoonup \mathcal{A}$
- A PAS is combinatorialy complete when it contains enough elements to represent all closed λ-terms (Formal definition given later)

Definition (Partial combinatory algebra – PCA)

A partial combinatory algebra (PCA) is a combinatorially complete PAS

Combinatorial completeness (1/3)

Let $\mathcal{A}$ be a partial applicative structure

Definition ($\mathcal{A}$-expressions)

Combinatory terms over $\mathcal{A}$ (or $\mathcal{A}$-expressions) are defined by:

$$\mathcal{A}\text{-expressions} \qquad t, u \quad ::= \quad x \quad | \quad a \quad | \quad tu \qquad (a \in \mathcal{A})$$

Syntactic worship: Free variables $FV(t)$, substitution $t[x := u]$

- **Remark:** Set of $\mathcal{A}$-expr. = free magma generated by $\mathcal{A} \uplus \mathrm{Var}$
- We define a (partial) interpretation function $t \mapsto t^{\mathcal{A}}$ from the set of closed $\mathcal{A}$-expressions to $\mathcal{A}$, using the inductive definition:

$$a^{\mathcal{A}} = a \qquad\qquad (tu)^{\mathcal{A}} = t^{\mathcal{A}} \cdot u^{\mathcal{A}}$$

- Notations:

$t\downarrow$	when	$t^{\mathcal{A}}$ is defined
$t\uparrow$	when	$t^{\mathcal{A}}$ is undefined
$t \cong u$	when	either $t, u\uparrow$ or $t, u\downarrow$ and $t^{\mathcal{A}} = u^{\mathcal{A}}$

Combinatorial completeness (2/3)

Definition (Combinatorial completeness)

A partial applicative structure $\mathcal{A}$ is combinatorially complete when for each $\mathcal{A}$-expression $t(x_1, \ldots, x_n)$ with free variables among $x_1, \ldots, x_n$ ($n \geq 1$), there exists $a \in \mathcal{A}$ such that for all $a_1, \ldots, a_n \in \mathcal{A}$:

1. $aa_1 \cdots a_{n-1} \downarrow$
2. $aa_1 \cdots a_n \cong t(a_1, \ldots, a_n)$

Notation: $a \;=\; (x_1, \ldots, x_n \mapsto t(x_1, \ldots, x_n))^{\mathcal{A}}$ (not unique, in general)

Theorem (Combinatorial completeness)

A partial applicative structure $\mathcal{A}$ is combinatorially complete iff there are two elements $\mathbf{K}, \mathbf{S} \in \mathcal{A}$ s.t. for all $a, b, c \in \mathcal{A}$:

1. $\mathbf{K}ab \downarrow$ and $\mathbf{K}ab = a$
2. $\mathbf{S}ab \downarrow$ and $\mathbf{S}abc \cong ac(bc)$

Combinatorial completeness (3/3)

- **Condition is necessary:** by combinatorial completeness, take

$$\mathbf{K} \;=\; (x,y \mapsto x)^{\mathcal{A}} \quad \text{and} \quad \mathbf{S} \;=\; (x,y,z \mapsto xz(yz))^{\mathcal{A}}$$

- To prove that condition is sufficient, use combinators $\mathbf{K}, \mathbf{S} \in \mathcal{A}$ to define λ-abstraction on the set of $\mathcal{A}$-expressions:

Definition of $\lambda x\,.\,t$:

$$\begin{array}{lcl@{\qquad}lcll} \lambda x\,.\,x & := & \mathbf{SKK} & \lambda x\,.\,y & := & \mathbf{K}\,y & \text{if } y \not\equiv x \\ \lambda x\,.\,a & := & \mathbf{K}\,a & \lambda x\,.\,tu & := & \mathbf{S}\,(\lambda x\,.\,t)\,(\lambda x\,.\,u) & \end{array}$$

By construction we have $FV(\lambda x\,.\,t) = FV(t) \setminus \{x\}$, and for each $\mathcal{A}$-expression $t(x)$ that depends (at most) on x:

$$\lambda x\,.\,t(x)\downarrow \qquad \text{and} \qquad (\lambda x\,.\,t(x))\,a \;\cong\; t(a) \quad \text{for all } a \in \mathcal{A}$$

- **Condition is sufficient:** if $\mathbf{K}, \mathbf{S} \in \mathcal{A}$ exist, put

$$(x_1, \ldots, x_n \mapsto t(x_1, \ldots, x_n))^{\mathcal{A}} \;:=\; (\lambda x_1 \cdots x_n\,.\,t(x_1, \ldots, x_n))^{\mathcal{A}}$$

Examples of partial combinatory algebras

Definition (Partial combinatory algebra – PCA)

A partial combinatory algebra (PCA) is a combinatorially complete PAS

- Examples of total combinatory algebras:
  - The set of closed λ-terms quotiented by β-conversion
  - The free magma generated by constants $\mathbf{K}, \mathbf{S}$ and quotiented by the relations $\mathbf{K}\,a\,b = a, \quad \mathbf{S}\,a\,b\,c = ac(bc)$ (Combinatory Logic)
- Examples of (really) partial combinatory algebras:
  - The set of closed λ-terms in normal form, equipped with the partial application defined by: $t \cdot u \;=\; \mathrm{NF}(tu)$
  - **Kleene's 1st model:** $\mathbb{N}$ equipped with $n \cdot p \;=\; \phi_n(p)$
  - **Kleene's 2d model:** based on $\mathbb{N}^{\mathbb{N}}$ + product topology
  - **The graph model:** based on $\mathfrak{P}(\omega)$ + product topology

Using partial combinatory algebras

- Using combinatory completeness, we can encode all constructs of system T in any partial combinatory algebra $\mathcal{A}$, for example:

  - $\texttt{pair} := (\lambda xyz\,.\,zxy)^{\mathcal{A}}$
  - $\pi_1 := (\lambda z\,.\,z\,(\lambda xy\,.\,x))^{\mathcal{A}}$
  - $\pi_2 := (\lambda z\,.\,z\,(\lambda xy\,.\,y))^{\mathcal{A}}$
  - $\texttt{0} := (\lambda xf\,.\,x)^{\mathcal{A}}$ $(= \mathbf{K})$
  - $\texttt{S} := (\lambda nxf\,.\,f\,n)^{\mathcal{A}}$ [Parigot]
  - $\mathbf{Y} := (\lambda f\,.\,(\lambda x\,.\,f\,(x\,x))\,(\lambda x\,.\,f\,(x\,x)))^{\mathcal{A}}$ [Church]
  - $\texttt{rec} := (\lambda x_0 x_1\,.\,\mathbf{Y}\,(\lambda rn\,.\,n\,x_0\,(\lambda z\,.\,x_1\,z\,(r\,z))))^{\mathcal{A}}$

- Using these constructions, we can define the relation or realizability $a \Vdash A$, where $a \in \mathcal{A}$ and A is a closed formula (exercise)

- Main Theorem holds in all PCA $\mathcal{A}$ (exercise), and depending on the choice of $\mathcal{A}$, we can realize more or less formulas

Where do the combinators $\mathbf{K}, \mathbf{S}$ come from?

- Through the CH correspondence, the types of combinators $\mathbf{K} = \lambda xy\,.\,x$ and $\mathbf{S} = \lambda xyz\,.\,xz(yz)$ correspond to the axioms of Hilbert deduction for minimal propositional logic:

$$\begin{array}{lcll} \mathbf{K} & = & \lambda xy\,.\,x & : \quad A \Rightarrow B \Rightarrow A \\ \mathbf{S} & = & \lambda xyz\,.\,xz(yz) & : \quad (A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow A \Rightarrow C \end{array}$$

Hilbert deduction for LJ

- **Rules:**

$$\frac{\vdash A \Rightarrow B \quad \vdash A}{\vdash B} \qquad \frac{\vdash A \Rightarrow B}{\vdash A \Rightarrow \forall x\, B}\ {\scriptstyle x \notin FV(A)} \qquad \frac{\vdash A \Rightarrow B}{\vdash \exists x\, A \Rightarrow B}\ {\scriptstyle x \notin FV(B)}$$

- **Axioms:**

$$A \Rightarrow B \Rightarrow A \qquad (A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow B) \Rightarrow A \Rightarrow C$$

$$A \Rightarrow B \Rightarrow A \wedge B \qquad A \wedge B \Rightarrow A \qquad A \wedge B \Rightarrow B \qquad \top \qquad \bot \Rightarrow A$$

$$A \Rightarrow A \vee B \qquad B \Rightarrow A \vee B \qquad (A \Rightarrow C) \Rightarrow (B \Rightarrow C) \Rightarrow A \vee B \Rightarrow C$$

$$\forall x\, A \Rightarrow A[x := e] \qquad A[x := e] \Rightarrow \exists x\, A$$

Extensions and variants

- **Extensions of Kleene realizability:**
  - To second- & higher-order Heyting arithmetic [Troelstra]
  - To intuitionistic & constructive set theories:
    - IZF_R, IZF_C [Myhill-Friedman 1973, McCarty 1984]
    - CZF [Aczel 1977]
- **Variants:**
  - Modified realizability [Kreisel]
  - Techniques of reducibility candidates [Tait, Girard, Parigot]
- **Categorical realizability:**
  - Strong connections with topoi [Scott, Hyland, Johnstone, Pitts]
- **Realizability for classical logic:**
  - Kleene realizability via a negative translation
  - Classical realizability in PA2, in ZF [Krivine 1994, 2001–]