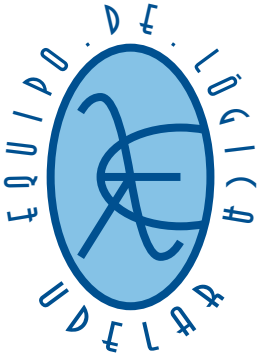
Computational interpretation of proofs: An introduction to realizability

Alexandre Miquel



Semantics of proofs and certified mathematics
PhD school – April 9th, 2014 – CIRM – Luminy

Typing versus realizability

Why does the term $\lambda x\,.\,x$ have type $\mathsf{nat} \to \mathsf{nat}$?

Typing

$$\dfrac{\dfrac{}{x : \mathsf{nat} \vdash x : \mathsf{nat}}}{\vdash \lambda x\,.\,x : \mathsf{nat} \to \mathsf{nat}}$$

- Syntactic analysis of terms
- At least semi-decidable
- Simple justification: derivation

Realizability

for all $n \in \mathsf{nat}$

$$(\lambda x\,.\,x)\,n \succ n \in \mathsf{nat}$$

- Computational analysis of terms
- Strongly undecidable
- External justification (proof)

Adequacy:

Correctness w.r.t. typing $\Rightarrow$ Correctness w.r.t. realizability

The hacker's point of view

Many situations where an ill-typed program is correct w.r.t. computation:

```
let my_stupid_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_stupid_function` always returns a Boolean when it is applied to an integer...

Two different notions of correction:

1. Correction w.r.t. typing
2. Correction w.r.t. computation $\rightsquigarrow$ Realizability

Plan

1. Realizability for Gödel's system T
2. Kleene realizability (with λ-terms)
3. Kreisel's modified realizability
4. Negative translations from LK to LJ

Plan

System T: common parts

Syntax

$$\begin{array}{lrcl} \textbf{Types} & A, B & ::= & \mathtt{nat} \mid A \times B \mid A \to B \\ \textbf{Terms} & M, N & ::= & x \mid \lambda x\,.\,M \mid MN \\ & & \mid & \mathtt{pair} \mid \mathtt{fst} \mid \mathtt{snd} \\ & & \mid & \mathtt{0} \mid \mathtt{S} \mid \mathtt{rec} \end{array}$$

Notations: $\langle M_1, M_2\rangle \equiv \mathtt{pair}\, M_1\, M_2$, $\quad \overline{n} \equiv \mathtt{S}^n\mathtt{0} \quad (n \in \mathbb{N})$

Reduction rules

$$\begin{array}{rcl} (\lambda x\,.\,M)N & \succ & M\{x := N\} \\ \mathtt{fst}\,\langle M_1, M_2\rangle & \succ & M_1 \\ \mathtt{snd}\,\langle M_1, M_2\rangle & \succ & M_2 \\ \mathtt{rec}\ M_0\ M_1\ \mathtt{0} & \succ & M_0 \\ \mathtt{rec}\ M_0\ M_1\ (\mathtt{S}\ N) & \succ & M_1\ N\ (\mathtt{rec}\ M_0\ M_1\ N) \end{array}$$

The point of view of typing (1/2)

Typing contexts $\Gamma \quad ::= \quad x_1 : A_1, \ldots, x_n : A_n \qquad (x_i \not\equiv x_j \text{ when } i \neq j)$

Typing rules

$$\frac{}{\Gamma \vdash x : A}\ {\scriptstyle (x:A)\in\Gamma}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x\,.\,M : A \to B} \qquad \frac{\Gamma \vdash M : A \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B}$$

$$\frac{}{\Gamma \vdash \mathtt{pair} : A \to B \to A \times B}$$

$$\frac{}{\Gamma \vdash \mathtt{fst} : A \times B \to A} \qquad \frac{}{\Gamma \vdash \mathtt{snd} : A \times B \to B}$$

$$\frac{}{\Gamma \vdash \mathtt{0} : \mathsf{nat}} \qquad \frac{}{\Gamma \vdash \mathtt{S} : \mathsf{nat} \to \mathsf{nat}}$$

$$\frac{}{\Gamma \vdash \mathtt{rec} : A \to (\mathsf{nat} \to A \to A) \to \mathsf{nat} \to A}$$

- **Problem:** Reduction is never mentioned in the rules!
  How to be sure that computation will not go wrong?

The point of view of typing (2/2)

Correction w.r.t. computation follows from 3 results:

Subject reduction (SR)

If $\Gamma \vdash M : A$ and $M \succ M'$, then $\Gamma \vdash M' : A$

Canonical forms of type nat (CF)

If $\vdash M : \mathrm{nat}$, M in normal form, then $M \equiv \overline{n}$ for some $n \in \mathbb{N}$

Strong normalization (SN)

If $\Gamma \vdash M : A$, then M is strongly normalizing

SR + CF + SN $\Rightarrow$ Every closed term $M : \mathrm{nat}$ reduces to a natural number $\overline{n}$

The point of view of realizability

Definition (Binary relation $M \Vdash A$, M closed)

1. $M \Vdash \mathrm{nat}$ if $M \succ^* \overline{n}$ for some $n \in \mathbb{N}$
2. $M \Vdash A \times B$ if $M \succ^* \langle M_1, M_2 \rangle$, where $M_1 \Vdash A$, $M_2 \Vdash B$
3. $M \Vdash A \to B$ if for all N: $N \Vdash A$ implies $MN \Vdash B$

- Closed terms: no typing context
- Purely computational definition: syntax = black box
- No correctness to prove: everything is in the definition!
- Requires an external justification: a proof (in which system?)
- Relation $M \Vdash A$ undecidable, not even semi-decidable

Lemma

For each A, the set $\{M \in \Lambda : M \Vdash A\}$ is closed under anti-reduction

Typing and realizability

Theorem (Adequacy)

If: $$x_1 : A_1, \ldots, x_n : A_n \vdash M : B$$

then for all $N_1 \Vdash A_1, \ldots, N_n \Vdash A_n$:

$$M\{x_1 := N_1, \ldots, x_n := N_n\} \Vdash B$$

Proof: By induction on the derivation, using the fact that each set $\{M \in \Lambda : M \Vdash A\}$ is closed under anti-reduction.

Particular cases (empty context)

- $\vdash M : A$ implies $M \Vdash A$
- $\vdash M : \mathrm{nat}$ implies $M \succ^* \overline{n}$ for some $n \in \mathbb{N}$

- **Remark:** In the previous proof of correctness (**SR** + **CF** + **SN**), a (customized) realizability model was hidden in the proof of **SN**

Plan

Background

Formalize the idea of constructivity according to Brouwer:

- 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*
- 1936. Kleene: *λ-definability and recursiveness*
- 1945. Kleene: *On the Interpretation of Intuitionistic Number Theory*

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
- Some unprovable formulas are realizable too...

Remarks:

- Codes for partial recursive functions can be replaced by the elements of any partial combinatory algebra
- Here, we shall take closed terms of Gödel's system T

Heyting Arithmetic

The language of HA

FO-terms $e, e' ::= x \mid f(e_1, \ldots, e_k)$

Formulas $A, B ::= e_1 = e_2 \mid \bot \mid A \Rightarrow B \mid A \wedge B \mid A \vee B \mid \forall x\, A \mid \exists x\, A$

- We assume given one function symbol f for each primitive recursive function: 0, S, $+$, $\times$, $\uparrow$, etc.
- For each closed FO-term e, write $[\![e]\!]$ its value $(\in \mathbb{N})$

Deduction rules and axioms

- Intuitionistic natural deduction: $A_1, \ldots, A_n \vdash B$
- Equality axioms (e.g. Leibniz axioms)
- Definitional axioms for primitive recursive functions
- Peano axioms: injectivity, non confusion, induction

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

Definition (realizability relation $M \Vdash A$)

$$
\begin{array}{lcl}
M \Vdash e_1 = e_2 & \equiv & [\![e_1]\!] = [\![e_2]\!] \ \wedge \ M \succ^* 0 \\
M \Vdash \bot & \equiv & \bot \\
M \Vdash A \wedge B & \equiv & \exists M_1, M_2 \ (M \succ^* \langle M_1, M_2 \rangle \ \wedge \ M_1 \Vdash A \ \wedge \ M_2 \Vdash B) \\
M \Vdash A \vee B & \equiv & \exists N \ (\ (M \succ^* \langle 0, N \rangle \ \wedge \ N \Vdash A) \ \vee \\
 & & \qquad (M \succ^* \langle 1, N \rangle \ \wedge \ N \Vdash B)\ ) \\
M \Vdash A \Rightarrow B & \equiv & \forall N \ (N \Vdash A \ \Rightarrow \ MN \Vdash B) \\
M \Vdash \forall x\, A(x) & \equiv & \forall n \ \ M\,\overline{n} \Vdash A(n) \\
M \Vdash \exists x\, A(x) & \equiv & \exists n \ \exists N \ (M \succ^* \langle \overline{n}, N \rangle \ \wedge \ N \Vdash A(n))
\end{array}
$$

Lemma

For each A, the set $\{M \in \Lambda : M \Vdash A\}$ is closed under anti-reduction

Adequacy

- Every FO-term e with free variables $x_1, \ldots, x_k$ is translated into a term e^* of system T with the same meaning (and free variables)
- Every derivation $d : (A_1, \ldots, A_n \vdash B)$ is translated into a term d^* of system T with free variables $x_1, \ldots, x_k, z_1, \ldots, z_n$, where:
  - $x_1, \ldots, x_k$ are the free variables of $A_1, \ldots, A_n, B$
  - $z_1, \ldots, z_n$ are proof variables associated to $A_1, \ldots, A_n$

  The construction of d^* follows the CH correspondence

Proposition (Adequacy)

Given a derivation $d : (A_1, \ldots, A_n \vdash B)$:

- for all valuations $\rho : \mathrm{Var} \to \mathbb{N}$
- for all realizers $N_1 \Vdash A_1[\rho], \ldots, N_n \Vdash A_n[\rho]$:

$$d^*[\rho]\{z_1 := N_1, \ldots, z_n := N_n\} \Vdash B[\rho]$$

Proof: By induction on d.

Extracting a term d^* from a valuation: some cases

$$\left(\frac{}{A_1, \ldots, A_n \vdash A_i}\right)^* = z_i \qquad \left(\begin{array}{c}\vdots\, d \\ \dfrac{\Gamma \vdash \bot}{\Gamma \vdash A}\end{array}\right)^* = \mathsf{any\_term}$$

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

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

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

Realizing the axioms of HA

- Equality axioms and defining equalities for primitive recursive functions have trivial realizers:

$$\begin{aligned} \lambda y\,.\,0 &\Vdash \forall y\ (0 + y = y) \\ \lambda xy\,.\,0 &\Vdash \forall x\ \forall y\ (s(x) + y = s(x + y)) \qquad \text{(etc.)} \end{aligned}$$

- Realizing Peano axioms:

$$\begin{aligned} \lambda xyz\,.\,z &\Vdash \forall x\ \forall y\ (s(x) = s(y) \Rightarrow x = y) \\ \text{any\_term} &\Vdash \forall x\,(s(x) \neq 0) \\ \texttt{rec} &\Vdash A(0) \ \Rightarrow\ \forall x\,(A(x) \Rightarrow A(s(x))) \ \Rightarrow\ \forall x\,A(x) \end{aligned}$$

Theorem (Kleene)

If $\mathrm{HA} \vdash A$, then $M \Vdash A$ for some M (A, M closed)

Optimizing realizers

- Many true or provable formulas can be given trivial realizers, without even looking at the proof:

Realizing true equalities

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

- Example: $\lambda xy\,.\,0 \Vdash \forall x\,\forall y\,(x + y = y + x)$
- The formula

$$\forall x\,\forall y\,\forall z\,\forall n\,(x \neq 0 \Rightarrow y \neq 0 \Rightarrow n > 2 \Rightarrow x^n + y^n \neq z^n)$$

has a realizer that fits into the margin[1]

- This can be generalized to all Harrop formulas (see later)

[1] E.g. $\lambda z\,.\,z$. For the proof that it is a realizer, see [Wiles-Taylor'95]

Unprovable, but realizable

- Consequence of the Halting problem:

$$\not\Vdash \ \forall x\,(\mathrm{Halt}(x) \ \vee \ \neg\mathrm{Halt}(x))$$

Corollary: HA $\not\vdash$ EM (from Adequacy)

- Therefore, the negation of the above formula is realized:

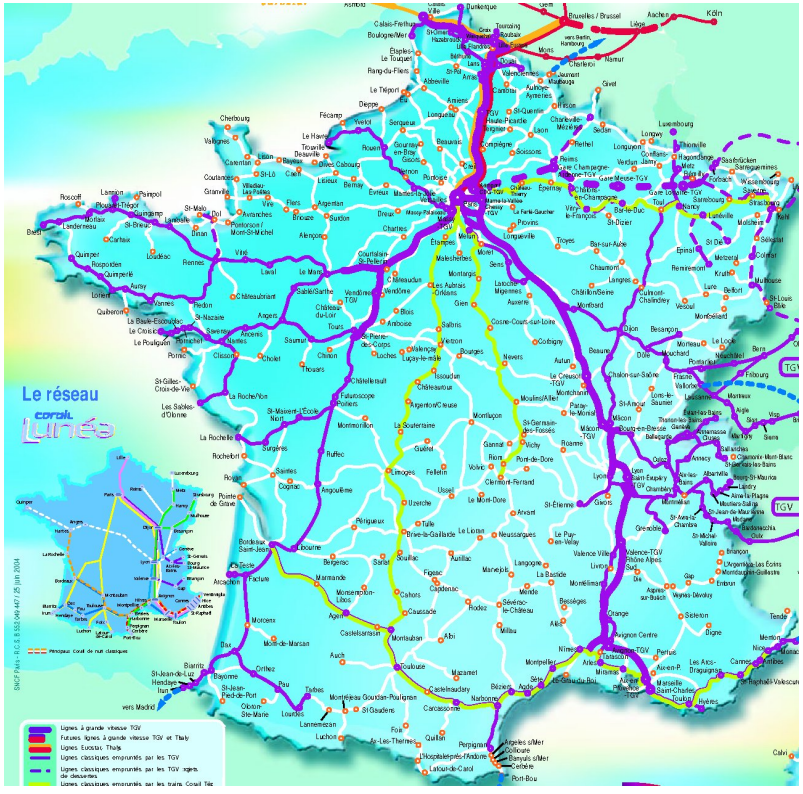
any_term $\Vdash \ \neg\forall x\,(\mathrm{Halt}(x) \ \vee \ \neg\mathrm{Halt}(x))$ (Realizable, but not provable)

- Markov Principle:

$$M \ \Vdash \ \forall x\ (P(x) \vee \neg P(x)) \ \Rightarrow \ \neg\forall x\,\neg P(x) \ \Rightarrow \ \exists x\,P(x)$$

where $M \ \equiv \ \lambda u\_\,.\,\boldsymbol{Y}\ (\lambda fn\,.\,\texttt{match}\ u\ n\ \texttt{with}$
$\mid \langle 0, p\rangle \ \mapsto \ \langle n, p\rangle$
$\mid \langle 1, \_\rangle \ \mapsto \ f\ (\mathrm{S}\ n))\ 0$

Realizability vs. provability/typing (metaphore)



Program extraction

Theorem (program extraction)

If $M \Vdash \forall x\, \exists y\, A(x,y)$, then for all $n \in \mathbb{N}$:

$$M\,\overline{n} \;\succ^*\; \langle \overline{p}, N \rangle$$

for some $p \in \mathbb{N}$ (witness) and $N \Vdash A(n,p)$ (justification)

- Extracted program: $f \equiv \lambda x\,.\,\mathtt{fst}\,(M\,x)$
- **Problem:** $N \Vdash A(n,p) \;\not\Rightarrow\; \mathsf{HA} \vdash A(n,p)$
- **Solution:** glued realizability

Glued realizability (1/2)

$\mathcal{P}$ set of closed formulas such that:

- $\mathcal{P}$ contains all theorems of HA
- $\mathcal{P}$ closed under modus ponens $\quad ((A \Rightarrow B) \in \mathcal{P},\ A \in \mathcal{P} \ \rightsquigarrow\ B \in \mathcal{P})$

$$
\begin{array}{lcl}
M \Vdash_{\mathcal{P}} n = m & \equiv & n = m \ \wedge\ M \succ^* 0 \\
M \Vdash_{\mathcal{P}} \bot & \equiv & \bot \\
M \Vdash_{\mathcal{P}} A \wedge B & \equiv & \exists M_1, M_2\ (M \succ^* \langle M_1, M_2 \rangle \ \wedge\ M_1 \Vdash_{\mathcal{P}} A \ \wedge\ M_2 \Vdash_{\mathcal{P}} B) \\
M \Vdash_{\mathcal{P}} A \vee B & \equiv & \exists N\ (\ (M \succ^* \langle 0, N \rangle \ \wedge\ N \Vdash_{\mathcal{P}} A) \ \vee \\
 & & \qquad (M \succ^* \langle 1, N \rangle \ \wedge\ N \Vdash_{\mathcal{P}} B)\ ) \\
M \Vdash_{\mathcal{P}} A \Rightarrow B & \equiv & \forall N\ (N \Vdash_{\mathcal{P}} A \ \Rightarrow\ MN \Vdash_{\mathcal{P}} B) \quad \wedge \quad (A \Rightarrow B) \in \mathcal{P} \\
M \Vdash_{\mathcal{P}} \forall x\, A(x) & \equiv & \forall n \ \ M\, n \Vdash_{\mathcal{P}} A(n) \quad \wedge \quad (\forall x\, A(x)) \in \mathcal{P} \\
M \Vdash_{\mathcal{P}} \exists x\, A(x) & \equiv & \exists n\ \exists M'\ (M \succ^* \langle n, M' \rangle \ \wedge\ M' \Vdash_{\mathcal{P}} A(n))
\end{array}
$$

Note: Plain realizability $=$ case where $\mathcal{P}$ contains all formulas

Glued realizability (2/2)

Theorem (Kleene)

1. If $M \Vdash_{\mathcal{P}} A$, then $A \in \mathcal{P}$
2. If $\mathrm{HA} \vdash A$, then there is a term M such that $M \Vdash_{\mathcal{P}} A$

- Case where $\mathcal{P}$ is the set of all closed formulas $\rightsquigarrow$ Plain realizability
- Case where $\mathcal{P}$ is the set of all theorems of HA:
  1. A provable in HA $\Leftrightarrow$ A is $\mathcal{P}$-realizable
  2. Disjunction property: $\mathrm{HA} \vdash A \vee B \rightsquigarrow \mathrm{HA} \vdash A$ or $\mathrm{HA} \vdash B$
  3. Witness property: $\mathrm{HA} \vdash \exists x\, A(x) \rightsquigarrow n + \mathrm{HA} \vdash A(n)$

(without using cut-elimination!)

Kleene's presentation

- Kleene did not used closed λ-terms, but Gödel codes for partial recursive functions, equipped with:
  - a recursive bijection $\langle \cdot, \cdot \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$
  - Kleene's (partial) application: $n \cdot m = \phi_n(m)$
    (where $(\phi_n)_{n \in \mathbb{N}}$ is an enumeration of all partial recursive functions)
- Convenient to formalize realizability within HA:

$$A \quad \mapsto \quad n \Vdash A \qquad \text{(formula translation)}$$

Theorem: If $\mathrm{HA} \vdash A$, then there is $n \in \mathbb{N}$ such that $\mathrm{HA} \vdash (n \Vdash A)$

- **Remark:** We can do the same with closed λ-terms, or with any partial combinatory algebra that is definable in HA.

Extensions and variants

- **Extensions:**
  - To second-order arithmetic
  - To intuitionistic set theories (IZ, IZF, CZF): Friedman, Myhill, McCarty, Aczel, etc.
- **Variants:**
  - Modified realizability: Kreisel
  - Techniques of reducibility candidates: Tait, Girard, Parigot, etc.
- **Categorical realizability:**
  - Strong connections with topos theory: Scott, Hyland, Johnstone, Pitts
- **Realizability for classical logic:**
  - Krivine realizability (in PA2, ZF)

Plan

The arithmetic of finite types: HA^ω

- Multi-sorted first-order logic whose sorts are the types of system T:

$$\tau, \sigma \quad ::= \quad \mathsf{bool} \quad | \quad \mathsf{nat} \quad | \quad \tau \times \sigma \quad | \quad \tau \to \sigma$$

- **Individuals of sort** $\tau \quad \equiv \quad$ terms of type τ (in system T)

- **Formulas:**

$$A, B \quad ::= \quad \mathtt{at}(M) \quad | \quad A \wedge B \quad | \quad A \Rightarrow B \quad | \quad \forall x^\tau A \quad | \quad \exists x^\tau A$$

  - $\mathtt{at}(M)$ means: $M = \mathtt{tt}$ $(M : \mathsf{bool})$
  - $\bot \equiv \mathtt{at}(\mathtt{ff})$
  - $M_1 =_{\mathsf{nat}} M_2 \equiv \mathtt{at}(\mathsf{nat\_eq}\, M_1\, M_2)$ $(M_1, M_2 : \mathsf{nat})$
  - $A \vee B \equiv \exists x^{\mathsf{bool}} ((x =_{\mathsf{bool}} \mathtt{tt} \Rightarrow A) \wedge (x =_{\mathsf{bool}} \mathtt{ff} \Rightarrow B))$

- **Axioms:** Equality, computation, Peano axioms

- **Remark:** This system is a conservative extension of HA
  (Not to be confused with higher-order arithmetic)

Kreisel's modified realizability

- To every formula A we associate a type A^* of system T:

$$\begin{array}{rcl@{\qquad}rcl} (\mathtt{at}(M))^* & \equiv & \mathsf{nat} & & & \\ (A \wedge B)^* & \equiv & A^* \times B^* & (\exists x^\tau\, A)^* & \equiv & \tau \times A^* \\ (A \Rightarrow B)^* & \equiv & A^* \to B^* & (\forall x^\tau\, A)^* & \equiv & \tau \to A^* \end{array}$$

- To every formula A we associate another formula z **mr** A
  (whose free variables are those of A, plus $z : A^*$)

$$\begin{array}{lcl} z\ \mathbf{mr}\ \mathtt{at}(M) & \equiv & z = 0\ \wedge\ \mathtt{at}(M) \\ z\ \mathbf{mr}\ A \wedge B & \equiv & \mathtt{fst}(z)\ \mathbf{mr}\ A\ \wedge\ \mathtt{snd}(z)\ \mathbf{mr}\ B \\ z\ \mathbf{mr}\ A \Rightarrow B & \equiv & \forall u\ (u\ \mathbf{mr}\ A\ \Rightarrow\ z\,u\ \mathbf{mr}\ B) \\ z\ \mathbf{mr}\ \forall x^\tau A(x) & \equiv & \forall x^\tau\ \ z\,x\ \mathbf{mr}\ A(x) \\ z\ \mathbf{mr}\ \exists x^\tau A(x) & \equiv & \mathtt{snd}(z)\ \mathbf{mr}\ A(\mathtt{fst}(z)) \end{array}$$

Theorem: If $\mathrm{HA}^\omega \vdash A$, then $\mathrm{HA}^\omega \vdash M\ \mathbf{mr}\ A$ for some $M : A^*$

Properties of Kreisel's modified realizability

- Allows to prove that: $\mathsf{HA} \nvdash \mathsf{Markov}$
- Allows to realize
  - The principle of Independence of Premises:

$$(\neg A \Rightarrow \exists x\, B(x)) \;\Rightarrow\; \exists x\, (\neg A \Rightarrow B(x))$$

  - The (type-theoretic) Axiom of Choice

$$\forall x^{\tau}\; \exists y^{\sigma} A(x,y) \;\Rightarrow\; \exists f^{\tau\to\sigma}\; \forall x^{\tau} A(x, f(x))$$

- **Practical interest:**
  - Program extraction towards a typed language (system T)
  - Possibility of optimizing extraction by erasing Harrop formulas:

$$H \quad ::= \quad \mathtt{at}(M) \quad | \quad H \wedge H \quad | \quad A \Rightarrow H \quad | \quad \forall x^{\tau}\, H$$

Optimizing modified realizers (1/2)

- We introduce a pseudo-type ϵ expressing computational irrelevance and optimize the definition of the type A^* as follows:

$$A^* \equiv \epsilon \quad (A \text{ atomic})$$

$$(A \wedge B)^* \equiv \begin{cases} \epsilon & \text{if } A^* \equiv B^* \equiv \epsilon \\ B^* & \text{if } A^* \equiv \epsilon,\ B^* \not\equiv \epsilon \\ A^* & \text{if } A^* \not\equiv \epsilon,\ B^* \equiv \epsilon \\ A^* \times B^* & \text{si } A^* \not\equiv \epsilon,\ B^* \not\equiv \epsilon \end{cases}$$

$$(A \Rightarrow B)^* \equiv \begin{cases} \epsilon & \text{if } B^* \equiv \epsilon \\ B^* & \text{if } A^* \equiv \epsilon,\ B^* \not\equiv \epsilon \\ A^* \to B^* & \text{si } A^* \not\equiv \epsilon,\ B^* \not\equiv \epsilon \end{cases}$$

$$(\forall x^\sigma A)^* \equiv \begin{cases} \epsilon & \text{if } A^* \equiv \epsilon \\ \sigma \to A^* & \text{if } A^* \not\equiv \epsilon \end{cases}$$

$$(\exists x^\sigma A)^* \equiv \begin{cases} \sigma & \text{if } A^* \equiv \epsilon \\ \sigma \times A^* & \text{if } A^* \not\equiv \epsilon \end{cases}$$

By construction: $A^* \equiv \epsilon$ iff A Harrop

Optimizing modified realizers (2/2)

- For every non-Harrop formula A (i.e. $A^* \not\equiv \epsilon$) we modify the definition of the relation z **mr** A accordingly (Exercise!)

  No modified realizability / program extraction for Harrop formulas

Theorem: If $\mathrm{HA}^\omega \vdash A$, then $\mathrm{HA}^\omega \vdash M$ **mr** A for some $M : A^*$

(Provided A is non-Harrop)

- **Example:** Euclidian division

$$\forall x^{\mathsf{nat}}\, \forall y^{\mathsf{nat}}\, (y \neq 0 \;\Rightarrow\; \exists q^{\mathsf{nat}}\, \exists r^{\mathsf{nat}}\, (x = qy + r \;\wedge\; r < y))$$

  - Associated type is: $\mathsf{nat} \to \mathsf{nat} \to \mathsf{nat} \times \mathsf{nat}$

    A modified realizer M **mr** A (extracted from a proof of A) will only compute the function of interest

  - Method used by Schwichtenberg et al. in **MinLog**

Plan

How to cope with classical logic?

- Kleene realizability is definitely incompatible with classical logic

$$\not\Vdash \quad \forall x\,(\mathrm{Halt}(x) \vee \neg \mathrm{Halt}(x))$$

$$\mathrm{any\_term} \quad \Vdash \quad \neg\forall x\,(\mathrm{Halt}(x) \vee \neg \mathrm{Halt}(x))$$

  The same holds for modified realizability

- Two possible solutions:

  1. Compose Kleene realizability with a negative translation from classical logic (LK) to intuitionistic logic (LJ) (next slide)
  2. Reformulate the principles of realizability to make them compatible with classical logic: Krivine's classical realizability (next lecture)

The Gödel-Gentzen translation

- **Idea:** Turn positive constructions (atomic formulas, $\vee$, $\exists$) into negative constructions ($\bot$, $\neg$, $\Rightarrow$, $\wedge$, $\forall$) using De Morgan laws
- Every formula A is translated into the formula A^G defined by:

$$\bot^G \equiv \bot \qquad (e_1 = e_2)^G \equiv \neg\neg(e_1 = e_2)$$

$$(A \Rightarrow B)^G \equiv A^G \Rightarrow B^G$$

$$(A \wedge B)^G \equiv A^G \wedge B^G \qquad (A \vee B)^G \equiv \neg(\neg A^G \wedge \neg B^G)$$

$$(\forall x\, A)^G \equiv \forall x\, A^G \qquad (\exists x\, A)^G \equiv \neg\forall x\, \neg A^G$$

writing: $\neg A \equiv A \Rightarrow \bot$

Theorem (Soundness)

1. $\mathsf{LK} \vdash A^G \Leftrightarrow A$
2. If $\mathsf{PA} \vdash A$, then $\mathsf{HA} \vdash A^G$

Realizing translated formulas

- **Strategy:**
  1. Build a derivation d of A (in PA)
  2. Turn it into a derivation d^G of A^G (in HA)
  3. Turn d^G into a Kleene realizer (program extraction)
- Does not work! Failure comes from:

Proposition (Realizability collapse)

For every formula A, Kleene's semantics for A^G mimics Tarski's semantics for A:

$$\{M \in \Lambda \ : \ M \Vdash A^G\} \ = \ \begin{cases} \Lambda & \text{if} \quad \mathbb{N} \models A \\ \varnothing & \text{if} \quad \mathbb{N} \not\models A \end{cases}$$

Proof. By induction on A.

- **Reason:** A^G is always a Harrop formula (no computational contents)
- **Conclusion:** Kleene $\circ$ Gödel-Gentzen $=$ Tarski

Friedman's R-translation

(called A-translation by Friedman)

- **Principle:** In Gödel-Gentzen's translation, replace each occurrence of $\bot$ (absurdity) by a fixed formula R, called the return formula
- Every formula A is translated into the formula A^F defined by:

$$\bot^F \equiv R \qquad (e_1 = e_2)^F \equiv \neg_R\neg_R(e_1 = e_2)$$

$$(A \Rightarrow B)^F \equiv A^F \Rightarrow B^F$$

$$(A \wedge B)^F \equiv A^F \wedge B^F \qquad (A \vee B)^F \equiv \neg_R(\neg_R A^F \wedge \neg_R B^F)$$

$$(\forall x\, A)^F \equiv \forall x\, A^F \qquad (\exists x\, A)^F \equiv \neg_R\forall x\, \neg_R A^F$$

writing: $\neg_R A \equiv A \Rightarrow R$

Theorem (Soundness)

If $\mathrm{PA} \vdash A$, then $\mathrm{HA} \vdash A^F$ (independently from the formula R)

Beware! The two formulas A and A^F are no more (classically) equivalent

Friedman's trick

Theorem (Kreisel-Friedman)

PA conservatively extends HA over Π^0_2-formulas:

If $\quad \mathrm{PA} \vdash \forall x\, \exists y\, f(x,y) = 0, \quad$ then $\quad \mathrm{HA} \vdash \forall x\, \exists y\, f(x,y) = 0$

Proof. Assume that $\mathrm{PA} \vdash \forall x\, \exists y\, f(x,y) = 0$. We have:

$\mathrm{HA} \vdash \forall x \neg_R \forall y \neg_R \neg_R \neg_R f(x,y) = 0$ (by R-translation)
$\mathrm{HA} \vdash \forall x \neg_R \forall y \neg_R f(x,y) = 0$ (since $\neg_R \neg_R \neg_R \Leftrightarrow \neg_R$)
$\mathrm{HA} \vdash \neg_R \forall y \neg_R f(x_0,y) = 0$ ($\forall$-elim, x_0 fresh)
$\mathrm{HA} \vdash \forall y\, (f(x_0,y) = 0 \Rightarrow R) \Rightarrow R$ (def. of $\neg_R$)

We now let: $R \equiv \exists y_0\, f(x_0,y_0) = 0$ (Friedman's trick!) From the def. of R:

$$\mathrm{HA} \vdash \forall y\, (f(x_0,y) = 0 \Rightarrow \exists y_0\, f(x_0,y_0) = 0) \Rightarrow \exists y_0\, f(x_0,y_0) = 0$$

But the premise of the above implication is provable

$$\mathrm{HA} \vdash \forall y\, (f(x_0,y) = 0 \Rightarrow \exists y_0\, f(x_0,y_0) = 0) \qquad (\exists\text{-intro})$$

hence we get

$\mathrm{HA} \vdash \exists y_0\, f(x_0,y_0) = 0$ (modus ponens)
$\mathrm{HA} \vdash \forall x_0\, \exists y_0\, f(x_0,y_0) = 0$ ($\forall$-intro) □

Realizing translated formulas, again

- **Strategy:**
  1. Build a derivation d of a Π^0_2-formula A (in PA)
  2. Turn it into a derivation F-trick(d^F) of A (in HA)
  3. Turn F-trick(d^F) into a Kleene realizer of A (program extraction)

- This technique perfectly works in practice. However:
  - The formula A^F is never a Harrop formula, even when A is.
    Possible fix: Introduce specific optimization techniques, e.g.:
    Refined Program Extraction [Berger et al., 2001]
  - The translation $A \mapsto A^F$ completely changes the structure of the underlying proof. **Possible fix:** cf next slides

The Lafont-Reus-Streicher translation (1/4)

- **Idea:** Translate each formula A into the (relative) negation of a formula $A^\perp$ representing the negation of A:

$$A^{LRS} \equiv \neg_R A^\perp \qquad (A^\perp \text{ defined by induction on } A)$$

(Again, this translation is parameterized by a return formula R)

- To every predicate symbol p (source language) we associate a predicate symbol $p^\perp$ representing its negation (target language)

- Definition of the translations $A \mapsto A^\perp$ and $A \mapsto A^{LRS}$:

$$\begin{array}{rclcrcl} (p(e_1,\ldots,e_k))^\perp & \equiv & p^\perp(e_1,\ldots,e_k) & & \perp^\perp & \equiv & \top \\ (A \Rightarrow B)^\perp & \equiv & A^{LRS} \wedge B^\perp & & (\forall x\, A)^\perp & \equiv & \exists x\, A^\perp \end{array}$$

$$A^{LRS} \equiv \neg_R A^\perp$$

Theorem (Soundness)

If $\mathsf{LK} \vdash A$, then $\mathsf{LJ} \vdash A^{LRS}$ (independently from the formula R)

The Lafont-Reus-Streicher translation (2/4)

- **Intuition:** The translated formula $A^{\perp}$ represents the type of stacks opposing (classical) terms of type A:

$$(A_1 \Rightarrow \cdots \Rightarrow A_n \Rightarrow B)^{\perp} \equiv A_1^{LRS} \wedge \cdots \wedge A_n^{LRS} \wedge B^{\perp}$$

$$(A_1 \to \cdots \to A_n \to B)^{\perp} \equiv A_1^{LRS} \times \cdots \times A_n^{LRS} \times B^{\perp}$$

- To analyze the computational contents of the LRS-translation, we need to work across two λ-calculi:
  - A source calculus to represent classical proofs:

    $\lambda_{\mathbf{source}} = \lambda_{\to} + \text{ɔc} : ((A \to B) \to A) \to A$ (Peirce's law)

    (Polymorphic constant ɔc introduces classical reasoning)
  - An intutionistic target calculus to represent translated proofs:

    $\lambda_{\mathbf{target}} = \lambda_{\to,\times}$

    (In this calculus, pairs are used to represent stacks)

The Lafont-Reus-Streicher translation (3/4)

- The underlying CPS-translation:

$$
\begin{array}{rcl}
(x)^{LRS} & \equiv & \lambda s\,.\,x\,s \\
(\lambda x\,.\,t)^{LRS} & \equiv & \lambda\langle x, s\rangle\,.\,t^{LRS}\,s \\
(tu)^{LRS} & \equiv & \lambda s\,.\,t^{LRS}\,\langle u^{LRS}, s\rangle \\
\text{cc}^{LRS} & \equiv & \lambda\langle x, s\rangle\,.\,x\,\langle \mathsf{k}_s, s\rangle\,, \\
\text{where}\ \ \mathsf{k}_s & \equiv & \lambda\langle y, \_\rangle\,.\,y\,s
\end{array}
$$

Theorem (Soundness)

If $\Gamma \vdash t : A$ (in the source λ-calculus)

then $\Gamma^{LRS} \vdash t^{LRS} : A^{LRS}$ (in the target λ-calculus)

The Lafont-Reus-Streicher translation (4/4)

- From the Lafont-Reus-Streicher translation...

$$\begin{array}{rcl}
(\lambda x\,.\,t)^{LRS} \mathbin{@} \langle u, s\rangle & \succ & t^{LRS}\{x := u\} \mathbin{@} s \\
(tu)^{LRS} \mathbin{@} s & \succ & t^{LRS} \mathbin{@} \langle u^{LRS}, s\rangle \\
\mathrm{cc}^{LRS} \mathbin{@} \langle u, s\rangle & \succ & u \mathbin{@} \langle k_s, s\rangle \\
\mathrm{k}_s \mathbin{@} \langle u, s'\rangle & \succ & u \mathbin{@} s
\end{array}$$

... to the Krivine Abstract Machine (KAM)

$$\begin{array}{lrcl}
\textbf{Grab} & \lambda x\,.\,t \star u \cdot \pi & \succ & t\{x := u\} \star \pi \\
\textbf{Push} & tu \star \pi & \succ & t \star u \cdot \pi \\
\textbf{Save} & \mathrm{cc} \star u \cdot \pi & \succ & u \star \mathrm{k}_\pi \cdot \pi \\
\textbf{Restore} & \mathrm{k}_\pi \star u \cdot \pi' & \succ & u \star \pi
\end{array}$$

- Reformulating Kleene realizability through the LRS-translation (and its CPS), we get Krivine's classical realizability (cf next lecture)