Synthetic Formalization of Posts Problem

Bachelor Seminar Talk

Felix Jahn

May 26, 2020

Advisor: Yannick Forster
Supervisor: Gert Smolka

SAARLAND UNIVERSITY

How to show Undecidability?

How to show Undecidability?

Q.: Show the Totality Problem undecidable.

A.: We reduce from H:

...

How to show Undecidability?

Q.: Show the Totality Problem undecidable.
A.: We reduce from H:

...

Q.: Show PCP undecidable.
A.: We reduce from H:

...

How to show Undecidability?

Q.: Show the Totality Problem undecidable.
A.: We reduce from H:

...

Q.: Show PCP undecidable.
A.: We reduce from H:

...

Q.: Show TSAT undecidable.
A.: We reduce from H:

...

How to show Undecidability?

Q.: Show the Totality Problem undecidable.
A.: We reduce from H:

...

Q.: Show PCP undecidable.
A.: We reduce from H:

...

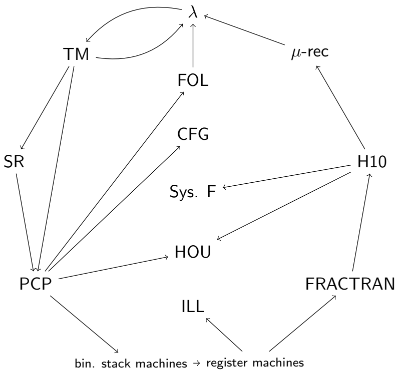
Q.: Show TSAT undecidable.
A.: We reduce from H:

...

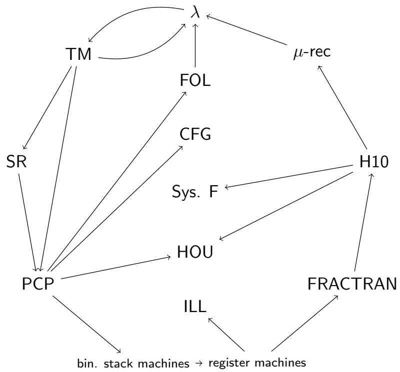
Q.: Show the Verification Problem undecidable.
A.: We reduce from H:

...

Undecidabilty Library



Undecidabilty Library



All enumerable but undecidable problems seem to reduce from the halting problem.

Posts Problem

RECURSIVELY ENUMERABLE SETS OF POSITIVE INTEGERS AND THEIR DECISION PROBLEMS

EMIL L. POST

Introduction. Recent developments of symbolic logic have considerable importance for mathematics both with respect to its philosophy and practice. That mathematicians generally are oblivious to

Figure: Posts Paper from 1944

Posts Problem for $\leq$

Is there an enumerable, but undecidable set P with $H \nleq P$?

Posts Problem

RECURSIVELY ENUMERABLE SETS OF POSITIVE INTEGERS AND THEIR DECISION PROBLEMS

EMIL L. POST

Introduction. Recent developments of symbolic logic have considerable importance for mathematics both with respect to its philosophy and practice. That mathematicians generally are oblivious to

Figure: Posts Paper from 1944

Posts Problem for $\leq$

Is there an enumerable, but undecidable set P with $H \not\leq P$?

- Many-one reduction $\leq_m$
- Simple sets, solving Posts Problem for $\leq_m$

Simple Set[1]

Definition (Traditional Simple Set)

A set $S \subseteq \mathbb{N}$ is called simple if it is enumerable, co-infinite, and contains an element from every enumerable, infinite set.

[1] originally by Post (1944), we follow the presentation by Rogers (1967)

Simple Set[1]

Definition (Traditional Simple Set)

A set $S \subseteq \mathbb{N}$ is called simple if it is enumerable, co-infinite, and contains an element from every enumerable, infinite set.

Remember the desired properties of a simple set S:

- S should be enumerable.
- S should be undecidable.
- H should not many-one reduce to S.

[1] originally by Post (1944), we follow the presentation by Rogers (1967)

Simple Set[1]

Definition (Traditional Simple Set)

A set $S \subseteq \mathbb{N}$ is called simple if it is enumerable, co-infinite, and contains an element from every enumerable, infinite set.

Remember the desired properties of a simple set S:

- S should be enumerable.
- S should be undecidable.
- H should not many-one reduce to S.

Definition (M-Completeness)

...

Definition (Creativeness)

...

Definition (Productiveness)

...

[1]originally by Post (1944), we follow the presentation by Rogers (1967)

Simple Set[1]

Definition (Traditional Simple Set)

A set $S \subseteq \mathbb{N}$ is called simple if it is enumerable, co-infinite, and contains an element from every enumerable, infinite set.

Remember the desired properties of a simple set S:

- S should be enumerable.
- S should be undecidable.
- H should not many-one reduce to S.

Theorem (Post)

There exists a simple set.

[1] originally by Post (1944), we follow the presentation by Rogers (1967)

Synthetic Approach[2]

Abstract from a concrete model of computation! Instead, take Coq as the model of computation:

[2] explored by Richman (1983) and Bauer (2006)

Synthetic Approach[2]

Abstract from a concrete model of computation! Instead, take Coq as the model of computation:

Definition (Decidability and Enumerability)

For a predicate $p : X \to \mathbb{P}$ we define:

$$\mathcal{D}\, p := \exists (f : X \to \mathbb{B}).\, \forall x. px \leftrightarrow fx = \mathsf{tt}$$
$$\mathcal{E}\, p := \exists (f : \mathbb{N} \to \mathbb{O}X).\, \forall x. px \leftrightarrow \exists n. fn = \mathsf{Some}\ x$$

Definition (Many-One Reduction)

For a predicates $p : X \to \mathbb{P}$ and $q : Y \to \mathbb{P}$ we define:

$$p \preceq_m q := \exists (f : X \to Y).\, \forall x. px \leftrightarrow q(fx)$$

[2] explored by Richman (1983) and Bauer (2006)

Synthetic Approach[3]: Axioms

Coq has for example no universal program. Therefore, we have to make some assumptions about the non-concrete model:

[3] explored by Richman (1983) and Bauer (2006)

Synthetic Approach[3]: Axioms

Coq has for example no universal program. Therefore, we have to make some assumptions about the non-concrete model:

- An enumerator $\mathcal{W} : \mathbb{N} \to (X \to \mathbb{P})$ for enumerable predicates:

$$\forall p.\, \mathcal{E}p \leftrightarrow (\exists c. \forall x. px \leftrightarrow \mathcal{W}cx)$$

"$\mathcal{W}cx$ iff program with index c halts on input x".

[3] explored by Richman (1983) and Bauer (2006)

Synthetic Approach[3]: Axioms

Coq has for example no universal program. Therefore, we have to make some assumptions about the non-concrete model:

- An enumerator $\mathcal{W} : \mathbb{N} \to (X \to \mathbb{P})$ for enumerable predicates:

  $$\forall p.\, \mathcal{E}p \leftrightarrow (\exists c. \forall x. px \leftrightarrow \mathcal{W}cx)$$

  "$\mathcal{W}cx$ iff program with index c halts on input x".

- The enumerability of $\mathcal{W}$.

[3] explored by Richman (1983) and Bauer (2006)

Synthetic Approach[3]: Axioms

Coq has for example no universal program. Therefore, we have to make some assumptions about the non-concrete model:

- An enumerator $\mathcal{W} : \mathbb{N} \to (X \to \mathbb{P})$ for enumerable predicates:
$$\forall p.\, \mathcal{E}p \leftrightarrow (\exists c.\forall x.px \leftrightarrow \mathcal{W}cx)$$
"$\mathcal{W}cx$ iff program with index c halts on input x".
- The enumerability of $\mathcal{W}$.
- The computability of the index of programs deciding finite predicates.

[3] explored by Richman (1983) and Bauer (2006)

Synthetic Approach[3]: Axioms

Coq has for example no universal program. Therefore, we have to make some assumptions about the non-concrete model:

- An enumerator $\mathcal{W} : \mathbb{N} \to (X \to \mathbb{P})$ for enumerable predicates:

  $$\forall p.\, \mathcal{E}p \leftrightarrow (\exists c. \forall x. px \leftrightarrow \mathcal{W}cx)$$

  "$\mathcal{W}cx$ iff program with index c halts on input x".

- The enumerability of $\mathcal{W}$.

- The computability of the index of programs deciding finite predicates.

- A corollary from the S-M-N-Theorem.

[3] explored by Richman (1983) and Bauer (2006)

Simple Predicate

Definition (Constructive Simple Predicate)

A predicate $p : X \to \mathbb{P}$ is called simple if it is enumerable, co-infinite and its complement contains no enumerable, infinite subset, e.g

$$\mathcal{E}\, p \land \text{infinite } \overline{p} \land \forall q : \text{infinite } q \to \mathcal{E}\, q \to q \not\subseteq \overline{p}.$$

Simple Predicate

Definition (Constructive Simple Predicate)

A predicate $p : X \to \mathbb{P}$ is called simple if it is enumerable, co-infinite and its complement contains no enumerable, infinite subset, e.g

$$\mathcal{E}\, p \land \text{infinite } \overline{p} \land \forall q : \text{infinite } q \to \mathcal{E}\, q \to q \nsubseteq \overline{p}.$$

Simple Predicate

Definition (Constructive Simple Predicate)

A predicate $p : X \to \mathbb{P}$ is called simple if it is enumerable, co-infinite and its complement contains no enumerable, infinite subset, e.g

$$\mathcal{E}\, p \land \text{infinite}\ \overline{p} \land \forall q : \text{infinite}\ q \to \mathcal{E}\, q \to q \nsubseteq \overline{p}.$$

Definition

$p : X \to \mathbb{P}$ is infinite, if there exists an injection $f : \mathbb{N} \to X$ with $Ran\ f \subseteq p$.

Simple Predicate

Definition (Constructive Simple Predicate)

A predicate $p : X \to \mathbb{P}$ is called simple if it is enumerable, co-infinite and its complement contains no enumerable, infinite subset, e.g

$$\mathcal{E}\, p \wedge \text{infinite}\ \overline{p} \wedge \forall q : \text{infinite}\ q \to \mathcal{E}\, q \to q \nsubseteq \overline{p}.$$

Definition

$p : X \to \mathbb{P}$ is infinite, if there exists an injection $f : \mathbb{N} \to X$ with $Ran\ f \subseteq p$.

$\overline{p}$ infinite via injection f

Simple Predicate

Definition (Constructive Simple Predicate)

A predicate $p : X \to \mathbb{P}$ is called simple if it is enumerable, co-infinite and its complement contains no enumerable, infinite subset, e.g

$$\mathcal{E}\, p \land \text{infinite } \overline{p} \land \forall q : \text{infinite } q \to \mathcal{E}\, q \to q \nsubseteq \overline{p}.$$

Definition

$p : X \to \mathbb{P}$ is infinite, if there exists an injection $f : \mathbb{N} \to X$ with $Ran\ f \subseteq p$.

$\overline{p}$ infinite via injection f

$\Rightarrow Ran\ f$ is an infinite and enumerable subset of $\overline{p}$.

Simple Predicate

Definition (Constructive Simple Predicate)

A predicate $p : X \to \mathbb{P}$ is called simple if it is enumerable, co-infinite and its complement contains no enumerable, infinite subset, e.g

$$\mathcal{E}\, p \land \text{infinite } \overline{p} \land \forall q : \text{infinite } q \to \mathcal{E}\, q \to q \nsubseteq \overline{p}.$$

Definition

$p : X \to \mathbb{P}$ is infinite, if there exists an injection $f : \mathbb{N} \to X$ with $Ran\ f \subseteq p$.

$\overline{p}$ infinite via injection f

$\Rightarrow Ran\ f$ is an infinite and enumerable subset of $\overline{p}$.

Simple Predicate

Definition (Constructive Simple Predicate)

A predicate $p : X \to \mathbb{P}$ is called simple if it is enumerable, co-infinite and its complement contains no enumerable, infinite subset, e.g

$$\mathcal{E}\, p \land \text{infinite } \overline{p} \land \forall q : \text{infinite } q \to \mathcal{E}\, q \to q \nsubseteq \overline{p}.$$

Definition

p is infinite, if it is not finite, e.g. $\neg \exists L. \forall x. px \to x \in L$.

Infinite criterias

Lemma

$p : X \to \mathbb{P}$ is infinite if for every n:

$$\exists L.\ |L| \geq n \land \mathsf{NoDup}\ L \land \forall x.\ x \in L \to p x$$

Infinite criterias

Lemma

$p : X \to \mathbb{P}$ is infinite if for every n:

$$\exists L.\ |L| \geq n \land \text{NoDup } L \land \forall x.\ x \in L \to p x$$

How do you show the existence of such a list?

$\rightsquigarrow$

Compute it!

Infinite criterias

Lemma

$p : X \to \mathbb{P}$ is infinite if for every n:

$$\neg\neg\ \exists L.\ |L| \geq n \land \text{NoDup } L \land \forall x.\ x \in L \to px$$

How do you show the existence of such a list?

⇝

Compute it!

$\Rightarrow$ Show the Double Negation!

Construction of a synthetic simple predicate

Remember the assumed enumerator for enumerable predicates:

$$\mathcal{W} : \mathbb{N} \to (\mathbb{N} \to \mathbb{P}) \text{ with } \forall p.\, \mathcal{E}p \leftrightarrow (\exists c. \forall x. px \leftrightarrow \mathcal{W}cx).$$

Construction of a synthetic simple predicate

Remember the assumed enumerator for enumerable predicates:

$$\mathcal{W} : \mathbb{N} \to (\mathbb{N} \to \mathbb{P}) \text{ with } \forall p.\, \mathcal{E}p \leftrightarrow (\exists c.\forall x.px \leftrightarrow \mathcal{W}cx).$$

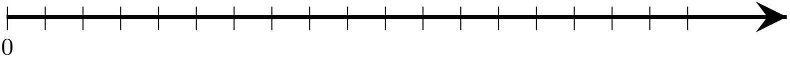
Consider the predicate: $C(x, y) := \mathcal{W}xy \wedge y > 2x$.
Defining S as $Ran\ C$?

Construction of a synthetic simple predicate

Remember the assumed enumerator for enumerable predicates:

$$\mathcal{W} : \mathbb{N} \to (\mathbb{N} \to \mathbb{P}) \text{ with } \forall p.\, \mathcal{E}p \leftrightarrow (\exists c. \forall x. px \leftrightarrow \mathcal{W}cx).$$

Consider the predicate: $C(x, y) := \mathcal{W}xy \wedge y > 2x$.
Defining S as $Ran\ C$?

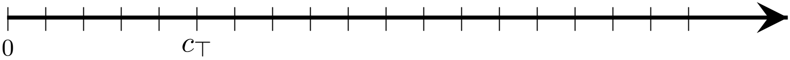


Construction of a synthetic simple predicate

Remember the assumed enumerator for enumerable predicates:

$$\mathcal{W} : \mathbb{N} \to (\mathbb{N} \to \mathbb{P}) \text{ with } \forall p.\, \mathcal{E}p \leftrightarrow (\exists c. \forall x. px \leftrightarrow \mathcal{W}cx).$$

Consider the predicate: $C(x, y) := \mathcal{W}xy \wedge y > 2x$.
Defining S as $Ran\ C$?

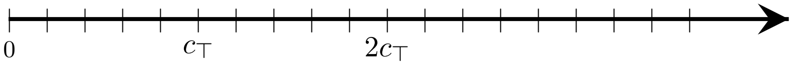


Construction of a synthetic simple predicate

Remember the assumed enumerator for enumerable predicates:

$$\mathcal{W} : \mathbb{N} \to (\mathbb{N} \to \mathbb{P}) \text{ with } \forall p.\, \mathcal{E}p \leftrightarrow (\exists c. \forall x. px \leftrightarrow \mathcal{W}cx).$$

Consider the predicate: $C(x, y) := \mathcal{W}xy \wedge y > 2x$.
Defining S as $Ran\ C$?

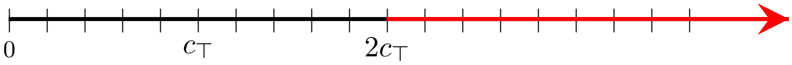


Construction of a synthetic simple predicate

Remember the assumed enumerator for enumerable predicates:

$$\mathcal{W} : \mathbb{N} \to (\mathbb{N} \to \mathbb{P}) \text{ with } \forall p.\, \mathcal{E}p \leftrightarrow (\exists c. \forall x. px \leftrightarrow \mathcal{W}cx).$$

Consider the predicate: $C(x, y) := \mathcal{W}xy \wedge y > 2x$.
Defining S as $Ran\ C$?

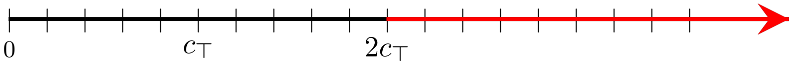


Construction of a synthetic simple predicate

Remember the assumed enumerator for enumerable predicates:

$$\mathcal{W} : \mathbb{N} \to (\mathbb{N} \to \mathbb{P}) \text{ with } \forall p.\, \mathcal{E}p \leftrightarrow (\exists c.\forall x.px \leftrightarrow \mathcal{W}cx).$$

Consider the predicate: $C(x, y) := \mathcal{W}xy \wedge y > 2x$.
Defining S as $Ran\ C$?



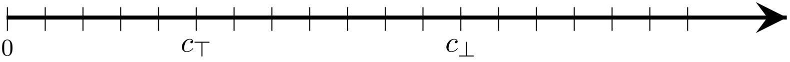
We need a mapping ψ with $C(x, \psi x)$.

Construction of a synthetic simple predicate

Remember the assumed enumerator for enumerable predicates:

$$\mathcal{W} : \mathbb{N} \to (\mathbb{N} \to \mathbb{P}) \text{ with } \forall p.\, \mathcal{E}p \leftrightarrow (\exists c. \forall x. px \leftrightarrow \mathcal{W}cx).$$

Consider the predicate: $C(x, y) := \mathcal{W}xy \land y > 2x$.
Defining S as $Ran\ C$?



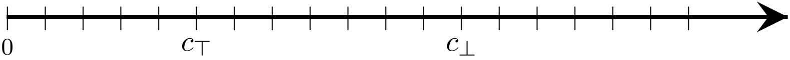
We need a mapping ψ with $C(x, \psi x)$.

Construction of a synthetic simple predicate

Remember the assumed enumerator for enumerable predicates:

$$\mathcal{W} : \mathbb{N} \to (\mathbb{N} \to \mathbb{P}) \text{ with } \forall p.\, \mathcal{E}p \leftrightarrow (\exists c. \forall x. px \leftrightarrow \mathcal{W}cx).$$

Consider the predicate: $C(x, y) := \mathcal{W}xy \wedge y > 2x$.
Defining S as $Ran\ C$?



We need a mapping ψ with $C(x, \psi x)$.
Unfortunately, ψ can't be total.

μ - Operator

If $p : \mathbb{N} \to \mathbb{P}$ is decidable, $\exists n.pn \to \Sigma n.pn$

μ - Operator

If $p : \mathbb{N} \to \mathbb{P}$ is decidable, $\exists n.pn \to \Sigma n.pn \land \forall y.py \to n \leq y$.
(Guarded minimisation operator $\mu_{\mathbb{N}}$)

μ - Operator

If $p : \mathbb{N} \to \mathbb{P}$ is decidable, $\exists n.pn \to \Sigma n.pn \wedge \forall y.py \to n \leq y$.
(Guarded minimisation operator $\mu_{\mathbb{N}}$)

Theorem ($\mu_{\mathcal{E}}$ - Operator)

For a enumerable predicate $p : Y \to \mathbb{P}$ with $\exists y.py$, we can compute a (unique) y with py by the $\mu_{\mathcal{E}}$ - Operator.

μ - Operator

If $p : \mathbb{N} \to \mathbb{P}$ is decidable, $\exists n.pn \to \Sigma n.pn \land \forall y.py \to n \leq y$.
(Guarded minimisation operator $\mu_{\mathbb{N}}$)

Theorem ($\mu_{\mathcal{E}}$ - Operator)

For a enumerable predicate $p : Y \to \mathbb{P}$ with $\exists y.py$, we can compute a (unique) y with py by the $\mu_{\mathcal{E}}$ - Operator.

Given x with $\exists y.C(x, y)$, define ψ using $\mu_{\mathcal{E}}$ for the enumerable predicate $\lambda y.C(x, y)$.

μ - Operator

If $p : \mathbb{N} \to \mathbb{P}$ is decidable, $\exists n.pn \to \Sigma n.pn \land \forall y.py \to n \leq y$.
(Guarded minimisation operator $\mu_{\mathbb{N}}$)

Theorem ($\mu_{\mathcal{E}}$ - Operator)

For a enumerable predicate $p : Y \to \mathbb{P}$ with $\exists y.py$, we can compute a (unique) y with py by the $\mu_{\mathcal{E}}$ - Operator.

Given x with $\exists y.C(x, y)$, define ψ using $\mu_{\mathcal{E}}$ for the enumerable predicate $\lambda y.C(x, y)$.

$$\Rightarrow \psi : \forall x. \left(\exists y.C(x, y)\right) \to \mathbb{N}$$

μ - Operator

If $p : \mathbb{N} \to \mathbb{P}$ is decidable, $\exists n.pn \to \Sigma n.pn \land \forall y.py \to n \leq y$.
(Guarded minimisation operator $\mu_{\mathbb{N}}$)

Theorem ($\mu_{\mathcal{E}}$ - Operator)

For a enumerable predicate $p : Y \to \mathbb{P}$ with $\exists y.py$, we can compute a (unique) y with py by the $\mu_{\mathcal{E}}$ - Operator.

Given x with $\exists y.C(x,y)$, define ψ using $\mu_{\mathcal{E}}$ for the enumerable predicate $\lambda y.C(x,y)$.

$$\Rightarrow \psi : \forall x. \left(\exists y.C(x,y)\right) \to \mathbb{N}$$

with $C(x, \psi x H)$.

Simple Predicate S

Definition

We define the simple predicate $S : \mathbb{N} \to \mathbb{P}$ as

$$Sy := \exists x. \exists H. \psi x H = y.$$

Simple Predicate S

Definition

We define the simple predicate $S : \mathbb{N} \to \mathbb{P}$ as

$$Sy := \exists x. \psi x = y.$$

Simple Predicate S

Definition

We define the simple predicate $S : \mathbb{N} \to \mathbb{P}$ as

$$Sy := \exists x. \psi x = y.$$

S should be simple and therefore

1. enumerable,
2. co-infinite,
3. $\overline{S}$ should not contain an enumerable and infinite subset.

Simple Predicate S

Definition

We define the simple predicate $S : \mathbb{N} \to \mathbb{P}$ as

$$Sy := \exists x. \psi x = y.$$

S should be simple and therefore

1. enumerable, ✓
2. co-infinite,
3. $\overline{S}$ should not contain an enumerable and infinite subset.

Simple Predicate S

Definition

We define the simple predicate $S : \mathbb{N} \to \mathbb{P}$ as

$$Sy := \exists x. \psi x = y.$$

S should be simple and therefore

1. enumerable, ✓
2. co-infinite,
3. $\overline{S}$ should not contain an enumerable and infinite subset. ✓

Co-Infinity

Lemma

For all $x : \mathbb{N}$: $C(x, \psi\, x)$ and therefore $\psi\, x > 2x$.

Co-Infinity

Lemma

For all $x : \mathbb{N}$: $C(x, \psi\, x)$ and therefore $\psi\, x > 2x$.

	$\psi\, 0$			$\psi\, 1$
0	1	2	3	4

Co-Infinity

Lemma

For all $x : \mathbb{N}$: $C(x, \psi\, x)$ and therefore $\psi\, x > 2x$.

	$\psi\, 0$			$\psi\, 1$
0	1	2	3	4

	$\psi\, 0$			$\psi\, 1$	$\psi\, 2$	
0	1	2	3	4	5	6

Co-Infinity

Lemma

For all $x : \mathbb{N}$: $C(x, \psi\, x)$ and therefore $\psi\, x > 2x$.

	$\psi\, 0$	$\dots$	$\psi(n{-}1)$

0 1 2 $\dots$ $2n-1$ $2n$

Co-Infinity

Lemma

For all $x : \mathbb{N}$: $C(x, \psi\, x)$ and therefore $\psi\, x > 2x$.

	$\psi\, 0$	$\dots$	$\psi(n{-}1)$

0 1 2 $\dots$ $2n-1$ $2n$

$\Rightarrow [0, 1, ..., 2n]$ contains at most n elements in S.

Co-Infinity

Lemma

For all $x : \mathbb{N}$: $C(x, \psi\, x)$ and therefore $\psi\, x > 2x$.

	$\psi\, 0$	$\dots$	$\psi(n{-}1)$

$0 \quad 1 \quad 2 \quad \dots \quad 2n-1 \quad 2n$

$\Rightarrow [0, 1, ..., 2n]$ contains at most n elements in S.

Definition

We say L lists p up to a bound b iff $\forall x. x \in L \leftrightarrow px \wedge x \leq b$.

Co-Infinity

Lemma

For all $x : \mathbb{N}$: $C(x, \psi\, x)$ and therefore $\psi\, x > 2x$.

	$\psi\, 0$	$\dots$	$\psi(n-1)$

0 1 2 ... $2n-1$ $2n$

$\Rightarrow [0, 1, ..., 2n]$ contains at most n elements in S.

Definition

We say L lists p up to a bound b iff $\forall x. x \in L \leftrightarrow p x \wedge x \leq b$.

$\Rightarrow$ If (duplicate free) L lists S up to $2n$: $|L| \leq n$

Co-Infinity

Definition

We say L lists p up to a bound b iff $\forall x. x \in L \leftrightarrow px \land x \leq b$.

Co-Infinity

Definition

We say L lists p up to a bound b iff $\forall x. x \in L \leftrightarrow px \wedge x \leq b$.

Lemma

We can show the "Non-Non Existence" of a list L that lists p up to b.

Co-Infinity

Definition

We say L lists p up to a bound b iff $\forall x. x \in L \leftrightarrow px \land x \leq b$.

Lemma

We can show the "Non-Non Existence" of a list L that lists p up to b.

Lemma

infinite p
$\leftrightarrow \forall n. \neg\neg\exists L. \ldots$

Co-Infinity

Definition

We say L lists p up to a bound b iff $\forall x. x \in L \leftrightarrow px \land x \leq b$.

Lemma

We can show the "Non-Non Existence" of a list L that lists p up to b.

Lemma

infinite p
$\leftrightarrow \forall n. \neg\neg\exists L. \dots$

Theorem

S is co-infinite.

Posts Problem

Theorem (Post)

There exists a simple predicate.

Posts Problem

Theorem (Post)

There exists a simple predicate.

→

Corollary

There exists an undecidable, but enumerable predicate S with $H \not\leq_m S$.

Many-One vs. One-One

"A one-one reduction is an injective many-one reduction"

Definition

$p \leq_1 q$ iff there is an injective function f, s.t. $px \leftrightarrow q(fx)$.

Many-One vs. One-One

"A one-one reduction is an injective many-one reduction"

Definition

$p \leq_1 q$ iff there is an injective function f, s.t. $px \leftrightarrow q(fx)$.

For a simple predicate S:

- $S \not\leq_1 S \times \mathbb{N}$ (Proof by notion of cylinder)
- $S \leq_m S \times \mathbb{N}$

Many-One vs. One-One

"A one-one reduction is an injective many-one reduction"

Definition

$p \leq_1 q$ iff there is an injective function f, s.t. $px \leftrightarrow q(fx)$.

Definition (Computability Degrees)

- $p \equiv_1 q := p \leq_1 q \land q \leq_1 p$
- $p \equiv_m q := p \leq_m q \land q \leq_m p$

For a simple predicate S:

- $S \not\leq_1 S \times \mathbb{N}$
- $S \leq_m S \times \mathbb{N}$

Many-One vs. One-One

"A one-one reduction is an injective many-one reduction"

Definition

$p \leq_1 q$ iff there is an injective function f, s.t. $px \leftrightarrow q(fx)$.

Definition (Computability Degrees)

- $p \equiv_1 q := p \leq_1 q \land q \leq_1 p$
- $p \equiv_m q := p \leq_m q \land q \leq_m p$

For a simple predicate S:

- $S \not\leq_1 S \times \mathbb{N}$
- $S \leq_m S \times \mathbb{N}$
- $S \not\equiv_1 S \times \mathbb{N}$
- $S \equiv_m S \times \mathbb{N}$

Conclusion

Theorem (Post)

There exists a simple predicate.

Corollary

There exists an undecidable, but enumerable predicate S with $H \not\leq_m S$.

Corollary

$\leq_m$ and $\leq_1$ just like $\equiv_m$ and $\equiv_1$ do not coincide on enumerable predicates.

Conclusion

Contributions:

- Synthetic approach for a formalization of:
  - Simple predicates
  - Posts Problem for $\leq_m$
  - Distinction of $\leq_1$ and $\leq_m$

  $\Rightarrow$ Complete mechanization in Coq ($\sim$ 2350 lines)
- Careful study of infinite predicates

Conclusion

Contributions:

- Synthetic approach for a formalization of:
  - Simple predicates
  - Posts Problem for $\leq_m$
  - Distinction of $\leq_1$ and $\leq_m$

  $\Rightarrow$ Complete mechanization in Coq ($\sim$ 2350 lines)
- Careful study of infinite predicates

Roadmap:

- Closer look at the synthetic axioms
- Myhills-Theorem
- Truth-table and Turing reduction, especially Posts Problem for these reductions

Backup Slides

Main References

- Emil Leon Post. 1944. Recursively enumerable sets of positive integers and their decision problems. *Bulletin of the American Mathematical Society* 50 (1944), 284–316.
- Hartley Rogers. 1967. The Theory of Recursive Functions and Effective Computability, *MIT Press*, 77-178.
- Nigel Cutland. 1980. Computability. *Cambridge University Press*, 121-142.
- Fred Richman. 1983. Church's thesis without tears. *The Journal of symbolic logic* 48, 3 (1983), 797–803.
- Andrej Bauer. 2006. First steps in synthetic computability theory. *Electronic Notes in Theoretical Computer Science* 155 (2006), 5–31.
- Yannick Forster and Dominik Kirst and Gert Smolka. 2019. On Synthetic Undecidability in Coq, with an Application to the Entscheidungsproblem. *8th ACM SIGPLAN International Conference on Certified Programs and Proofs*, 38-51.

Coq Development

	spec	proof
Preliminaries	415	748
Infinity	75	184
Posts Problem	353	568
TOTAL	843	1500

Synthetic Axioms

In the construction of S:

- An enumerator $\mathcal{W} : \mathbb{N} \to (X \to \mathbb{P})$ for enumerable predicates:

$$\forall p.\, \mathcal{E}p \leftrightarrow (\exists c.\forall x.px \leftrightarrow \mathcal{W}cx).$$

- Enumerability of $\mathcal{W}$.

In the proofs of the simple predicate properties:

- The computability of the program-index deciding finite predicates:

$$\Sigma\mathcal{C}.\forall nL.(\forall x.\mathcal{W}nx \leftrightarrow x \in L) \\ \to \forall mx.\mathcal{W}(\mathcal{C}mn)x \leftrightarrow x \in (m :: L).$$

- Corollary from S-M-N:

$$\forall f.\exists g.\forall nx.\mathcal{W}(gn)x \leftrightarrow \mathcal{W}n(fx).$$

Definition (M-Completeness)

A predicate $p : X \to \mathbb{P}$ is m-complete if it is enumerable and for all datatypes Y and all predicates $q : Y \to \mathbb{P}$, $\mathcal{E}q \to q \leq_m p$.

Definition (Productiveness)

A predicate $p : X \to \mathbb{P}$ is productive if there is a function $g : \mathbb{N} \to X$ with

$$\forall n. \mathcal{W}n \subseteq p \to p(gn) \land \neg \mathcal{W}n(gn).$$

Definition (Creativeness)

A predicate $p : X \to \mathbb{P}$ is creative if it is enumerable and its complement is productive.

Co-Infinity

Definition

We say L lists p up to a bound b iff $\forall x . x \in L \leftrightarrow p x \wedge x \leq b$.

Co-Infinity

Definition

We say L lists p up to a bound b iff $\forall x. x \in L \leftrightarrow px \wedge x \leq b$.

Lemma

We can show the "Non-Non Existence" of a list L that lists p up to b.

Lemma

infinite p
$\Leftrightarrow \forall n. \neg\neg\exists L. \ldots$

Co-Infinity

Definition

We say L lists p up to a bound b iff $\forall x. x \in L \leftrightarrow px \wedge x \leq b$.

Lemma

We can show the "Non-Non Existence" of a list L that lists p up to b.

Lemma

infinite p
$\Leftrightarrow \forall n. \neg\neg\exists L. \ldots$

If L lists S up to $2n$:

- $[0, ..., 2n] \backslash L$ lists $\overline{S}$ up to $2n$.

Co-Infinity

Definition

We say L lists p up to a bound b iff $\forall x. x \in L \leftrightarrow px \wedge x \leq b$.

Lemma

We can show the "Non-Non Existence" of a list L that lists p up to b.

Lemma

infinite p
$\Leftrightarrow \forall n. \neg\neg\exists L. \ldots$

If L lists S up to $2n$:

- $[0, ..., 2n] \backslash L$ lists $\overline{S}$ up to $2n$.
- $|L| \leq n$
- $|[0, ..., 2n] \backslash L| \geq n$

Co-Infinity

Definition

We say L lists p up to a bound b iff $\forall x. x \in L \leftrightarrow px \land x \leq b$.

Lemma

We can show the "Non-Non Existence" of a list L that lists p up to b.

Lemma

infinite p
$\Leftrightarrow \forall n. \neg\neg\exists L. \ldots$

If L lists S up to $2n$:

- $[0, ..., 2n] \backslash L$ lists $\overline{S}$ up to $2n$.
- $|L| \leq n$
- $|[0, ..., 2n] \backslash L| \geq n$

Theorem

S is co-infinite.

Definition (Cylinder)

A predicate $p : X \to \mathbb{P}$ is a cylinder, if there exists an isomorph type Y and $q : Y \to Prop$ with $p \equiv_1 q \times (\lambda y.\top)$.

Theorem

$$\text{cylinder } p \leftrightarrow p \equiv_1 p \times (\lambda x.\top)$$

One-One Reducibility

"A one-one reduction is an injective many-one reduction"

Definition

$p \leq_1 q$ iff there is an injective function f, s.t. $px \leftrightarrow q(fx)$.

Interesting properties:

- $p \leq_1 q \Rightarrow p \leq_m q$
- $p \leq_m q \Leftrightarrow p \times \mathbb{N} \leq_1 q \times \mathbb{N}$ and therefore
- $p \times \mathbb{N} \leq_m q \times \mathbb{N} \Leftrightarrow p \times \mathbb{N} \leq_1 q \times \mathbb{N}$

But do $\leq_1$ and $\leq_m$ coincide on all predicates?

Many-One vs. One-One

For a simple predicate S:

- $S \not\leq_1 S \times \mathbb{N}$ (proof by notion of cylinder)
- $S \leq_m S \times \mathbb{N}$ via $\lambda x.(x, 0)$ and $S \times \mathbb{N} \leq_m S$ via $\lambda(x, n).x$.

Definition

We define computability degrees:

- $p \equiv_1 q := p \leq_1 q \wedge q \leq_1 p$
- $p \equiv_m q := p \leq_m q \wedge q \leq_m p$

Clearly $p \equiv_1 q$ implies $p \equiv_m q$, but

$$S \not\equiv_1 S \times \mathbb{N} \text{ and } S \equiv_m S \times \mathbb{N}.$$