Gödel's incompleteness theorems

1 Properties of proof systems

$S = (\Sigma, L, A, R) \quad , S'(= \Sigma, L', A', R')$

def: extension We say S' *extends* S and write $S \subseteq S'$ if

$$\Sigma \subseteq \Sigma' \ , \ L \subseteq L' \ , \ A \subseteq A' \ , \ R \subseteq R'$$

Lemma 1. *If $S \subseteq S'$ and $S \vdash w$ then $S' \vdash w$.*

1 Properties of proof systems

$S = (\Sigma, L, A, R) \quad , S'(= \Sigma, L', A', R')$

def: extension We say S' *extends* S and write $S \subseteq S'$ if

$$\Sigma \subseteq \Sigma' \ , \ L \subseteq L' \ , \ A \subseteq A' \ , \ R \subseteq R'$$

Lemma 1. *If $S \subseteq S'$ and $S \vdash w$ then $S' \vdash w$.*

def: soundness/correctness Proof system S is *sound/correct* if all provable statements are true, i.e. if for all $w \in L$:

$$S \vdash w \Rightarrow w \text{ is true}$$

1 Properties of proof systems

$S = (\Sigma, L, A, R) \quad , S'(= \Sigma, L', A', R')$

def: extension We say S' *extends* S and write $S \subseteq S'$ if

$$\Sigma \subseteq \Sigma' \ , \ L \subseteq L' \ , \ A \subseteq A' \ , \ R \subseteq R'$$

Lemma 1. *If $S \subseteq S'$ and $S \vdash w$ then $S' \vdash w$.*

def: soundness/correctness Proof system S is *sound/correct* if all provable statements are true, i.e. if for all $w \in L$:

$$S \vdash w \Rightarrow w \text{ is true}$$

def:consistency an approximation to correctness. Proof system is *consistent* if for no statement w

$$S \vdash w \text{ and } S \vdash (\sim w)$$

1 Properties of proof systems

$S = (\Sigma, L, A, R) \quad , S'(= \Sigma, L', A', R')$

def: extension We say S' *extends* S and write $S \subseteq S'$ if

$$\Sigma \subseteq \Sigma' \; , \; L \subseteq L' \; , \; A \subseteq A' \; , \; R \subseteq R'$$

Lemma 1. *If $S \subseteq S'$ and $S \vdash w$ then $S' \vdash w$.*

def: soundness/correctness Proof system S is *sound/correct* if all provable statements are true, i.e. if for all $w \in L$:

$$S \vdash w \Rightarrow w \text{ is true}$$

def:consistency an approximation to correctness. Proof system is *consistent* if for no statement w

$$S \vdash w \text{ and } S \vdash (\sim w)$$

def: completeness Proof system S is *complete* if for all statements $w \in L$ either w or $\sim w$ is provable

$$S \vdash w \quad \text{or} \quad S \vdash (\sim w)$$

1 Properties of proof systems

$S = (\Sigma, L, A, R) \quad , S'(= \Sigma, L', A', R')$

def: extension We say S' *extends* S and write $S \subseteq S'$ if

$$\Sigma \subseteq \Sigma' \text{ , } L \subseteq L' \text{ , } A \subseteq A' \text{ , } R \subseteq R'$$

Lemma 1. *If $S \subseteq S'$ and $S \vdash w$ then $S' \vdash w$.*

def: soundness/correctness Proof system S is *sound/correct* if all proven predicates are true, i.e. if for all $w \in L$:

$$S \vdash w \Rightarrow w \text{ is true}$$

def:consistency an approximation to correctness. Proof system is *consistent* if for no predicate w

$$S \vdash w \text{ and } S \vdash (\sim w)$$

def: completeness Proof system S is *complete* if for all predicates $w \in L$ either w or $\sim w$ is provable

$$S \vdash w \quad \text{or} \quad S \vdash (\sim w)$$

Lemma 2. *If S is complete, then the language*

$$\{w \in L \ : \ S \vdash w\}$$

is decidable

1 Properties of proof systems

$S = (\Sigma, L, A, R) \quad , S'(= \Sigma, L', A', R')$

def: extension We say S' *extends* S and write $S \subseteq S'$ if

$$\Sigma \subseteq \Sigma' \; , \; L \subseteq L' \; , \; A \subseteq A' \; , \; R \subseteq R'$$

Lemma 1. *If $S \subseteq S'$ and $S \vdash w$ then $S' \vdash w$.*

def: soundness/correctness Proof system S is *sound/correct* if all proven predicates are true, i.e. if for all $w \in L$:

$$S \vdash w \Rightarrow w \text{ is true}$$

def:consistency an approximation to correctness. Proof system is *consistent* if for no predicate w

$$S \vdash w \text{ and } S \vdash (\sim w)$$

def: completeness Proof system S is *complete* if for all predicates $w \in L$ either w or $\sim w$ is provable

$$S \vdash w \quad \text{or} \quad S \vdash (\sim w)$$

Lemma 2. *If S is complete, then the language*

$$\{w \in L \; : \; S \vdash w\}$$

is decidable

Proof. Enumerate all proofs. If proof of w appears, accept. If proof of $\sim w$ appears, reject. $\square$

1 Properties of proof systems

$S = (\Sigma, L, A, R) \quad , S'(= \Sigma, L', A', R')$

def: extension We say S' *extends* S and write $S \subseteq S'$ if

$$\Sigma \subseteq \Sigma' \ , \ L \subseteq L' \ , \ A \subseteq A' \ , \ R \subseteq R'$$

Lemma 1. *If $S \subseteq S'$ and $S \vdash w$ then $S' \vdash w$.*

def: soundness/correctness Proof system S is *sound/correct* if all proven predicates are true, i.e. if for all $w \in L$:

$$S \vdash w \Rightarrow w \text{ is true}$$

def:consistency an approximation to correctness. Proof system is *consistent* if for no predicate w

$$S \vdash w \text{ and } S \vdash (\sim w)$$

def: completeness Proof system S is *complete* if for all predicates $w \in L$ either w or $\sim w$ is provable

$$S \vdash w \quad \text{or} \quad S \vdash (\sim w)$$

Lemma 2. *If S is complete, then the language*

$$\{w \in L \ : \ S \vdash w\}$$

is decidable

Proof. Enumerate all proofs. If proof of w appears, accept. If proof of $\sim w$ appears, reject. □

Lemma 3. *If S is correct and complete, then the set*

$$T = \{w \in L \ : \ w \text{ is true}\}$$

why?

is decidable

2 First Incompleteness Theorem

Lemma 4. *Let S be any proof system whose language is as expressive as L_Z*

$$S = (\Sigma_Z, L_Z, A, R)$$

If S is sound, then it is incomplete.

- reduce Halting problem H to T with lemma 5
- if Z_E would be complete, then by lemma 3 T would be decidable

Lemma 5. *There is a total computable function*

$$H : \{0,1,\#\} \to L_E$$

such that

$$H(u\#v) \text{ true} \Leftrightarrow \text{ TM } M_u \text{ started with } v \text{ halts}$$

2 First Incompleteness Theorem

Lemma 4. *Let S be any proof system whose language is as expressive as L_Z*

$$S = (\Sigma_Z, L_Z, A, R)$$

If S is sound, then it is incomplete.

- reduce Halting problem H to T with lemma 5
- if Z_E would be complete, then by lemma 3 T would be decidable

Lemma 5. *There is a total computable function*

$$H : \{0,1,\#\} \rightarrow L_E$$

such that

$$H(u\#v) \textit{ true} \Leftrightarrow \textit{TM } M_u \textit{ started with } v \textit{ halts}$$

from Sipser (MIT): Introduction to the Theory of Computation. A GREAT book. 'the actual construction of ... is too complicated to present here'.

- in contrast (KIU):

2 First Incompleteness Theorem

Lemma 5. *There is a total computable function*

$$H : \{0,1,\#\} \rightarrow L_E$$

such that

$$H(u\#v) \textit{ true} \Leftrightarrow \textit{ TM } M_u \textit{ started with } v \textit{ halts}$$

Consider 1-tape TM

$$M_u = (Z, A, \delta, z_0, E)$$

and $v \in \mathbb{B}^*$. M_u started with v halts iff

$$\exists w.\ w = \$k_0\$\ldots\$k_t\$ \quad , \quad w \in (A \cup Z \cup \{\$\})^+$$

with

1. $k_0 = B\ldots Bz_0vB\ldots B$
2. $k_i \vdash k_{i+1}$ for $i < t$
3. $|k_i| = |k_j|$ for all i, j
4. in no k_i if a $z \in Z$ first or last element
5. k_t is endconfiguraion

coding strings in numbers Let

$$p > \#A + \#Z + 2 \quad \text{prime number}$$

Interpret $w \in (A \cup Z \cup \{\$\})^*$ as number representation to base p.

- $$\psi : A \cup Z \cup \{\$\} \to [1 : p-1]$$
  $$\psi(a) \in \mathbb{N} \quad \text{injective}$$

- $$\psi(\varepsilon) = 0$$

coding strings in numbers Let

$$p > \#A + \#Z + 2 \quad \text{prime number}$$

Interpret $w \in (A \cup Z \cup \{\$\})^*$ as number representation to base p.

- $$\psi : A \cup Z \cup \{\$\} \to [1 : p-1]$$
  $$\psi(a) \in \mathbb{N} \quad \text{injective}$$
- $$\psi(\varepsilon) = 0$$
- extend to
  $$\psi : (A \cup Z \cup \{\$\})^* \to [1 : p-1]$$
  $$\psi(w[s-1:0]) = \sum_{i=1}^{s-1} \psi(w_i) \cdot p^i$$
- expression whose value codes $a \in A \cup Z \cup \{\$\}$
  $$\hat{a} = \overline{\psi(a)} \in T$$

coding strings in numbers Let

$$p > \#A + \#Z + 2 \quad \text{prime number}$$

Interpret $w \in (A \cup Z \cup \{\$\})^*$ as number representation to base p.

- $$\psi : A \cup Z \cup \{\$\} \to [1 : p-1]$$
  $$\psi(a) \in \mathbb{N} \quad \text{injective}$$

- $$\psi(\varepsilon) = 0$$

- extend to
  $$\psi : (A \cup Z \cup \{\$\})^* \to [1 : p-1]$$
  $$\psi(w[s-1:0]) = \sum_{i=1}^{s-1} \psi(w_i) \cdot p^i$$

- expression whose value codes $a \in A \cup Z \cup \{\$\}$
  $$\hat{a} = \overline{\psi(a)} \in T$$

expressing concatenation

Lemma 6. *There is an arithmetic predicate $v = [x,y]$ such that for all $V, X, Y \in (A \cup Z \cup \{\$\})^*$ holds*

$$\psi(V) = [\psi(X), \psi(Y)] \Leftrightarrow V = XY$$

v=xy

coding strings in numbers Let

$$p > \#A + \#Z + 2 \quad \text{prime number}$$

Interpret $w \in (A \cup Z \cup \{\$\})^*$ as number representation to base p.

- $$\psi : A \cup Z \cup \{\$\} \rightarrow [1 : p-1]$$
  $$\psi(a) \in \mathbb{N} \quad \text{injective}$$

- $$\psi(\varepsilon) = 0$$

- extend to
  $$\psi : (A \cup Z \cup \{\$\})^* \rightarrow [1 : p-1]$$
  $$\psi(w[s-1:0]) = \sum_{i=1}^{s-1} \psi(w_i) \cdot p^i$$

- expression whose value codes $a \in A \cup Z \cup \{\$\}$
  $$\hat{a} = \overline{\psi(a)} \in T$$

expressing concatenation

Lemma 6. *There is an arithmetic predicate $v = [x,y]$ such that for all $V, X, Y \in (A \cup Z \cup \{\$\})^*$ holds*

$$\psi(V) = [\psi(X), \psi(Y)] \Leftrightarrow V = XY$$

obviously using decomposition lemma for base p numbers. Top down:

- $$\begin{aligned} v = [x,y] \quad :\equiv \quad & (y = 0 \wedge v = x) \vee \\ & (\sim y = 0 \wedge \exists u \, (v = y + u \cdot x \wedge u = p^{|y|}) \end{aligned}$$

coding strings in numbers Let

$$p > \#A + \#Z + 2 \quad \text{prime number}$$

Interpret $w \in (A \cup Z \cup \{\$\})^*$ as number representation to base p.

- $$\psi : A \cup Z \cup \{\$\} \to [1 : p-1]$$
  $$\psi(a) \in \mathbb{N} \quad \text{injective}$$
- $$\psi(\varepsilon) = 0$$
- extend to
  $$\psi : (A \cup Z \cup \{\$\})^* \to [1 : p-1]$$
  $$\psi(w[s-1:0]) = \sum_{i=1}^{s-1} \psi(w_i) \cdot p^i$$
- expression whose value codes $a \in A \cup Z \cup \{\$\}$
  $$\hat{a} = \overline{\psi(a)} \in T$$

expressing concatenation

Lemma 6. *There is an arithmetic predicate $v = [x,y]$ such that for all $V, X, Y \in (A \cup Z \cup \{\$\})^*$ holds*

$$\psi(V) = [\psi(X), \psi(Y)] \Leftrightarrow V = XY$$

obviously using decomposition lemma for base p numbers. Top down:

- $$\begin{aligned} v = [x,y] \quad :\equiv \quad & (y = 0 \wedge v = x) \vee \\ & (\sim y = 0 \wedge \exists u \, (v = y + u \cdot x \wedge u = p^{|y|}) \end{aligned}$$
- $$\begin{aligned} u = p^{|y|} \quad :\equiv \quad & u \text{ is power of } p \\ & \wedge y < u \wedge u \le p \cdot u \end{aligned}$$

example:

$$456 < 1000 < 10 \cdot 456$$

coding strings in numbers Let

$$p > \#A + \#Z + 2 \quad \text{prime number}$$

Interpret $w \in (A \cup Z \cup \{\$\})^*$ as number representation to base p.

- $$\psi : A \cup Z \cup \{\$\} \to [1 : p-1]$$
  $$\psi(a) \in \mathbb{N} \quad \text{injective}$$

- $$\psi(\varepsilon) = 0$$

- extend to
  $$\psi : (A \cup Z \cup \{\$\})^* \to [1 : p-1]$$
  $$\psi(w[s-1:0]) = \sum_{i=1}^{s-1} \psi(w_i) \cdot p^i$$

- expression whose value codes $a \in A \cup Z \cup \{\$\}$
  $$\hat{a} = \overline{\psi(a)} \in T$$

expressing concatenation

Lemma 6. *There is an arithmetic predicate $v = [x,y]$ such that for all $V, X, Y \in (A \cup Z \cup \{\$\})^*$ holds*

$$\psi(V) = [\psi(X), \psi(Y)] \Leftrightarrow V = XY$$

obviously using decomposition lemma for base p numbers. Top down:

- $$\begin{aligned} v = [x,y] \quad :\equiv \quad & (y = 0 \wedge v = x) \vee \\ & (\sim y = 0 \wedge \exists u\, (v = y + u \cdot x \wedge u = p^{|y|}) \end{aligned}$$

- $$\begin{aligned} u = p^{|y|} \quad :\equiv \quad & u \text{ is power of } p \\ & \wedge\, y < u \wedge u \leq p \cdot u \end{aligned}$$

  example:
  $$456 < 1000 < 10 \cdot 456$$

- $$\begin{aligned} u \leq y \quad & :\equiv \quad \exists r\; u = y + r \\ u < y \quad & :\equiv \quad u \leq y \wedge \sim u = y \end{aligned}$$

coding strings in numbers Let

$$p > \#A + \#Z + 2 \quad \text{prime number}$$

Interpret $w \in (A \cup Z \cup \{\$\})^*$ as number representation to base p.

- $$\psi : A \cup Z \cup \{\$\} \to [1 : p-1]$$
  $$\psi(a) \in \mathbb{N} \quad \text{injective}$$

- $$\psi(\varepsilon) = 0$$

- extend to
  $$\psi : (A \cup Z \cup \{\$\})^* \to [1 : p-1]$$
  $$\psi(w[s-1:0]) = \sum_{i=1}^{s-1} \psi(w_i) \cdot p^i$$

- expression whose value codes $a \in A \cup Z \cup \{\$\}$
  $$\hat{a} = \overline{\psi(a)} \in T$$

expressing concatenation

Lemma 6. *There is an arithmetic predicate $v = [x,y]$ such that for all $V, X, Y \in (A \cup Z \cup \{\$\})^*$ holds*

$$\psi(V) = [\psi(X), \psi(Y)] \Leftrightarrow V = XY$$

obviously using decomposition lemma for base p numbers. Top down:

- $$\begin{aligned} v = [x,y] \quad :\equiv \quad & (y = 0 \wedge v = x) \vee \\ & (\sim y = 0 \wedge \exists u\, (v = y + u \cdot x \wedge u = p^{|y|}) \end{aligned}$$

- $$\begin{aligned} u = p^{|y|} \quad :\equiv \quad & u \text{ is power of } p \\ & \wedge y < u \wedge u \leq p \cdot u \end{aligned}$$

  example:
  $$456 < 1000 < 10 \cdot 456$$

- $$\begin{aligned} u \leq y \quad &:\equiv \quad \exists r\, u = y + r \\ u < v \quad &:\equiv \quad u < v \wedge \sim u = v \end{aligned}$$

- $$\begin{aligned} u \text{ is power of } p \quad &:\equiv \quad \forall r\, \forall s\, (r \cdot s = u \to r = 1 \vee p|r) \\ p|r \quad &:\equiv \quad \exists s\, r = \overline{p} \cdot s \end{aligned}$$

coding strings in numbers Let

$$p > \#A + \#Z + 2 \quad \text{prime number}$$

Interpret $w \in (A \cup Z \cup \{\$\})^*$ as number representation to base p.

- $$\psi : A \cup Z \cup \{\$\} \to [1 : p-1]$$
  $$\psi(a) \in \mathbb{N} \quad \text{injective}$$

- $$\psi(\varepsilon) = 0$$

- extend to
  $$\psi : (A \cup Z \cup \{\$\})^* \to [1 : p-1]$$
  $$\psi(w[s-1:0]) = \sum_{i=1}^{s-1} \psi(w_i) \cdot p^i$$

- expression whose value codes $a \in A \cup Z \cup \{\$\}$
  $$\hat{a} = \overline{\psi(a)} \in T$$

expressing concatenation

Lemma 6. *There is an arithmetic predicate $v = [x,y]$ such that for all $V, X, Y \in (A \cup Z \cup \{\$\})^*$ holds*

$$\psi(V) = [\psi(X), \psi(Y)] \Leftrightarrow V = XY$$

obviously using decomposition lemma for base p numbers. Top down:

- $$\begin{aligned} v = [x,y] \quad :\equiv \quad & (y = 0 \wedge v = x) \vee \\ & (\sim y = 0 \wedge \exists u \, (v = y + u \cdot x \wedge u = p^{|y|}) \end{aligned}$$

- $$\begin{aligned} u = p^{|y|} \quad :\equiv \quad & u \text{ is power of } p \\ & \wedge y < u \wedge u \leq p \cdot u \end{aligned}$$

  example:
  $$456 < 1000 < 10 \cdot 456$$

- $$\begin{aligned} u \leq y \quad & :\equiv \quad \exists r \, u = y + r \\ u < v \quad & :\equiv \quad u < v \wedge \sim u = v \end{aligned}$$

- $$\begin{aligned} u \text{ is power of } p \quad & :\equiv \quad \forall r \, \forall s \, (r \cdot s = u \to r = 1 \vee p|r) \\ p|r \quad & :\equiv \quad \exists s \, r = \overline{p} \cdot s \end{aligned}$$

concatenating several strings

$$u = [xyv] \; :\equiv \; \exists w \, (w = [x,y] \wedge u = [w,v])$$

etc.

Consider 1-tape TM

$$M_u = (Z, A, \delta, z_0, E)$$

and $v \in \mathbb{B}^*$. M_u started with v halts iff

$$\exists w.\ w = \$k_0\$\ldots\$k_t\$ \quad , \quad w \in (A \cup Z \cup \{\$\})^+$$

with

1. $k_0 = B\ldots Bz_0vB\ldots B$
2. $k_i \vdash k_{i+1}$ for $i < t$
3. $|k_i| = |k_j|$ for all i, j
4. in no k_i if a $z \in Z$ first or last element
5. k_t is endconfiguraion

(1): start configuration of computation

- $$\begin{array}{rcl} & & \exists k\ \exists m \exists a\ \exists b \exists c \\ (w & = & [\hat{\$}, k, \hat{\$}, m] \\ & & \wedge\ k = [a, b, c] \\ & & \wedge\ a = \psi(B\ldots B) \\ & & \wedge\ c = \psi(B\ldots B) \\ & & b = \psi(z_0 v_{n-1}\ldots v_0)) \end{array}$$

Consider 1-tape TM

$$M_u = (Z, A, \delta, z_0, E)$$

and $v \in \mathbb{B}^*$. M_u started with v halts iff

$$\exists w.\ w = \$k_0\$\ldots\$k_t\$ \quad , \quad w \in (A \cup Z \cup \{\$\})^+$$

with

1. $k_0 = B\ldots Bz_0vB\ldots B$
2. $k_i \vdash k_{i+1}$ for $i < t$
3. $|k_i| = |k_j|$ for all i, j
4. in no k_i if a $z \in Z$ first or last element
5. k_t is endconfiguraion

(1): start configuration of computation

- $$\begin{aligned} & \exists k\ \exists m \exists a\ \exists b \exists c \\ (w \quad = \quad & [\hat{\$}, k, \hat{\$}, m] \\ & \wedge\ k = [a, b, c] \\ & \wedge\ a = \psi(B\ldots B) \\ & \wedge\ c = \psi(B\ldots B) \\ & b = \psi(z_0 v_{n-1}\ldots v_0)) \end{aligned}$$

- $\sigma(d,a)$: d codes a single symbol in $\psi^{-1}(a)$

$$\sigma(d,a) :\equiv \exists e\ \exists f\ (a = [e,d,f]\ \wedge d < \overline{p} \wedge\ \sim d = 0)$$

Consider 1-tape TM

$$M_u = (Z, A, \delta, z_0, E)$$

and $v \in \mathbb{B}^*$. M_u started with v halts iff

$$\exists w.\ w = \$k_0\$\ldots\$k_t\$ \quad , \quad w \in (A \cup Z \cup \{\$\})^+$$

with

1. $k_0 = B\ldots B z_0 v B\ldots B$
2. $k_i \vdash k_{i+1}$ for $i < t$
3. $|k_i| = |k_j|$ for all i, j
4. in no k_i if a $z \in Z$ first or last element
5. k_t is endconfiguraion

(1): start configuration of computation

- $$\begin{aligned} & \exists k\ \exists m \exists a\ \exists b \exists c \\ (w \quad = \quad & [\hat{\$}, k, \hat{\$}, m] \\ & \wedge\ k = [a, b, c] \\ & \wedge\ a = \psi(B\ldots B) \\ & \wedge\ c = \psi(B\ldots B) \\ & b = \psi(z_0 v_{n-1}\ldots v_0)) \end{aligned}$$

- $\sigma(d,a)$: d codes a single symbol in $\psi^{-1}(a)$

$$\sigma(d,a) :\equiv \exists e\ \exists f\ (a = [e,d,f]\ \wedge d < \overline{p} \wedge\ \sim d = 0)$$

- $$a = \psi(B\ldots B) \quad :\equiv \forall d\ (\sigma(d,a) \rightarrow d = \hat{B})$$

Consider 1-tape TM

$$M_u = (Z, A, \delta, z_0, E)$$

and $v \in \mathbb{B}^*$. M_u started with v halts iff

$$\exists w.\ w = \$k_0\$\ldots\$k_t\$ \quad , \quad w \in (A \cup Z \cup \{\$\})^+$$

with

1. $k_0 = B\ldots Bz_0vB\ldots B$
2. $k_i \vdash k_{i+1}$ for $i < t$
3. $|k_i| = |k_j|$ for all i, j
4. in no k_i if a $z \in Z$ first or last element
5. k_t is endconfiguraion

(1): start configuration of computation

-
$$\begin{array}{rcl} & & \exists k\ \exists m \exists a\ \exists b \exists c \\ (w & = & [\hat{\$}, k, \hat{\$}, m] \\ & & \wedge\ k = [a, b, c] \\ & & \wedge\ a = \psi(B\ldots B) \\ & & \wedge\ c = \psi(B\ldots B) \\ & & b = \psi(z_0 v_{n-1}\ldots v_0)) \end{array}$$

- $\sigma(d, a)$: d codes a single symbol in $\psi^{-1}(a)$

$$\sigma(d,a) :\equiv \exists e\ \exists f\ (a = [e, d, f]\ \wedge d < \overline{p} \wedge\ \sim d = 0)$$

-
$$a = \psi(B\ldots B) \quad :\equiv \forall d\ (\sigma(d, a) \rightarrow d = \hat{B})$$

- Horner scheme for $\psi(z_0 v)$

$$\begin{array}{rcl} b = \psi(z_0 v) & :\equiv & \exists y_0 \ldots \exists y_s \\ (y_s & = & \widehat{z_0}\ \wedge \\ y_{s-1} & = & \overline{p} \cdot y_s + \widehat{v_{s-1}}\ \wedge \\ & \ldots & \\ y_0 & = & \overline{p} \cdot y_1 + \widehat{v_0}\ \wedge \\ b & = & y_0) \end{array}$$

Consider 1-tape TM

$$M_u = (Z, A, \delta, z_0, E)$$

and $v \in \mathbb{B}^*$. M_u started with v halts iff

$$\exists w.\ w = \$k_0\$\ldots\$k_t\$ \quad , \quad w \in (A \cup Z \cup \{\$\})^+$$

with

1. $k_0 = B\ldots Bz_0vB\ldots B$
2. $k_i \vdash k_{i+1}$ for $i < t$
3. $|k_i| = |k_j|$ for all i, j
4. in no k_i if a $z \in Z$ first or last element
5. k_t is endconfiguraion

(2) $\wedge$ (3): successor configuration

- in (codes of) neighboring configurations x and y we have $'x \vdash y'$

$$\begin{aligned} \forall r \forall x\ \forall y\ \forall z\ ((w &= [r, \hat{\$}, x, \hat{\$}y, \hat{\$}, z] \wedge {'x \text{ codes no } \$'} \wedge {'y \text{ codes no } \$'}) \\ &\to {'x \vdash y'}) \end{aligned}$$

Consider 1-tape TM

$$M_u = (Z, A, \delta, z_0, E)$$

and $v \in \mathbb{B}^*$. M_u started with v halts iff

$$\exists w.\ w = \$k_0\$\ldots\$k_t\$ \quad , \quad w \in (A \cup Z \cup \{\$\})^+$$

with

1. $k_0 = B\ldots Bz_0vB\ldots B$
2. $k_i \vdash k_{i+1}$ for $i < t$
3. $|k_i| = |k_j|$ for all i, j
4. in no k_i if a $z \in Z$ first or last element
5. k_t is endconfiguraion

•

$$'x \text{ codes no } \$' \;:\equiv\quad \forall d\ (\sigma(d,x) \rightarrow \sim d = \hat{\$})$$

(2) ∧ (3): successor configuration

- in (codes of) neighboring configurations x and y we have $'x \vdash y'$

$$\begin{aligned}\forall r \forall x\ \forall y\ \forall z\ ((w &= [r, \hat{\$}, x, \hat{\$}y, \hat{\$}, z] \wedge {}'x \text{ codes no } \$' \wedge {}'y \text{ codes no } \$') \\ &\rightarrow {}'x \vdash y')\end{aligned}$$

Consider 1-tape TM

$$M_u = (Z, A, \delta, z_0, E)$$

and $v \in \mathbb{B}^*$. M_u started with v halts iff

$$\exists w.\ w = \$k_0\$\ldots\$k_t\$ \quad , \quad w \in (A \cup Z \cup \{\$\})^+$$

with

1. $k_0 = B\ldots Bz_0vB\ldots B$
2. $k_i \vdash k_{i+1}$ for $i < t$
3. $|k_i| = |k_j|$ for all i, j
4. in no k_i if a $z \in Z$ first or last element
5. k_t is endconfiguraion

(2) $\wedge$ (3): successor configuration

- in (codes of) neighboring configurations x and y we have $'x \vdash y'$

$$\forall r \forall x\ \forall y\ \forall z\ ((w = [r, \hat{\$}, x, \hat{\$}y, \hat{\$}, z] \wedge\ 'x \text{ codes no } \$' \wedge\ 'y \text{ codes no } \$') \rightarrow\ 'x \vdash y')$$

- $$'x \text{ codes no } \$' \ :\equiv \quad \forall d\ (\sigma(d, x) \rightarrow \sim d = \hat{\$})$$

- next configuration

$$'x \vdash y' \ :\equiv \ \exists u\ \exists v$$
$$\Big(\bigvee_{z,q \in Z,\ a,c \in A\ ,\delta(z,a)=(q,c,N)} x = [u, \hat{z}, \hat{a}, v]\ \wedge\ y = [u, \hat{q}, \hat{c}, v]$$
$$\vee \bigvee_{z,q \in Z,\ a,c \in A\ ,\delta(z,a)=(q,c,R)} x = [u, \hat{z}, \hat{a}, v]\ \wedge\ y = [u, \hat{c}, \hat{q}, v]$$
$$\vee \bigvee_{z,q \in Z,\ a,b,c \in A\ ,\delta(z,a)=(q,c,L)} x = [u, \hat{b}, \hat{z}, \hat{a}, v]\ \wedge\ y = [u, \hat{q}, \hat{b}, \hat{c}, v]\Big)$$

Consider 1-tape TM

$$M_u = (Z, A, \delta, z_0, E)$$

and $v \in \mathbb{B}^*$. M_u started with v halts iff

$$\exists w.\ w = \$k_0\$\ldots\$k_t\$ \quad , \quad w \in (A \cup Z \cup \{\$\})^+$$

with

1. $k_0 = B\ldots Bz_0vB\ldots B$
2. $k_i \vdash k_{i+1}$ for $i < t$
3. $|k_i| = |k_j|$ for all i, j
4. in no k_i if a $z \in Z$ first or last element
5. k_t is endconfiguraion

(4): no state at border

$$\begin{aligned} & \forall x\, \forall y\, \forall v \\ ((w \quad = \quad & [x, \hat{\$}, y, \hat{\$}, v] \wedge {}'y \text{ codes no } \$' \\ & \wedge \forall d\, (\sigma(d, y) \wedge (\bigvee_{z \in Z} d = \hat{z}))) \\ \rightarrow \quad & \exists e\, \exists f\, (y = [e, d, f] \wedge \sim e = 0 \wedge \sim f = 0)) \end{aligned}$$

Consider 1-tape TM

$$M_u = (Z, A, \delta, z_0, E)$$

and $v \in \mathbb{B}^*$. M_u started with v halts iff

$$\exists w.\ w = \$k_0\$\ldots\$k_t\$ \quad , \quad w \in (A \cup Z \cup \{\$\})^+$$

with

1. $k_0 = B\ldots Bz_0vB\ldots B$
2. $k_i \vdash k_{i+1}$ for $i < t$
3. $|k_i| = |k_j|$ for all i, j
4. in no k_i if a $z \in Z$ first or last element
5. k_t is endconfiguraion

(4): no state at border

$$\begin{array}{rcl} & & \forall x\, \forall y\, \forall v \\ ((w & = & [x, \hat{\$}, y, \hat{\$}, v] \wedge \text{'}y \text{ codes no } \$\text{'} \\ & & \wedge \forall d\, (\sigma(d, y) \wedge (\bigvee_{z \in Z} d = \hat{z}))) \\ & \rightarrow & \exists e\, \exists f\, (y = [e, d, f] \wedge \sim e = 0 \wedge \sim f = 0)) \end{array}$$

(5) end state is reached

$$\exists e\, \exists f\, \exists t\ w = [e, t, f] \wedge (\bigvee_{z \in E} t = \hat{z})$$

Consider 1-tape TM

$$M_u = (Z, A, \delta, z_0, E)$$

and $v \in \mathbb{B}^*$. M_u started with v halts iff

$$\exists w.\ w = \$k_0\$\ldots\$k_t\$ \quad , \quad w \in (A \cup Z \cup \{\$\})^+$$

with

1. $k_0 = B\ldots Bz_0vB\ldots B$
2. $k_i \vdash k_{i+1}$ for $i < t$
3. $|k_i| = |k_j|$ for all i, j
4. in no k_i if a $z \in Z$ first or last element
5. k_t is endconfiguraion

(4): no state at border

$$\begin{array}{rcl} & & \forall x\ \forall y\ \forall v \\ ((w & = & [x, \hat{\$}, y, \hat{\$}, v] \wedge {}'y \text{ codes no } \$' \\ & & \wedge \forall d\ (\sigma(d, y) \wedge (\bigvee_{z \in Z} d = \hat{z}))) \\ & \rightarrow & \exists e\ \exists f\ (y = [e, d, f] \wedge \sim e = 0 \wedge \sim f = 0)) \end{array}$$

(5) end state is reached

$$\exists e\ \exists f\ \exists t\ w = [e, t, f] \wedge (\bigvee_{z \in E} t = \hat{z})$$

reduction:

$$H(u\#v) :\equiv\ \exists w\ (1) \wedge (2) \wedge (3) \wedge (4) \wedge (5)$$

3 Towards the second incompleteness theorem

3.1 Preliminaries

For proof systems S denote by $consis(S)$ the statement, that S is consistent.

Lemma 7. *Consistency can be formulated in* Z_E

$$consis(S) \in L_E$$

3 Towards the second incompleteness theorem

3.1 Preliminaries

For proof systems S denote by $consis(S)$ the statement, that S is consistent.

Lemma 7. *Consistency can be formulated in* Z_E

$$consis(S) \in L_E$$

Proof. Machine M_m started with empty tape enumerates all proofs of S and records all proven statements p. If it finds proofs of p and $\sim p$ for some p it halts. Then

$$consis(S) :\equiv H(m\#)$$

□

3 Towards the second incompleteness theorem

3.1 Preliminaries

For proof systems S denote by $consis(S)$ the statement, that S is consistent.

Lemma 7. *Consistency can be formulated in Z_E*

$$consis(S) \in L_E$$

Proof. Machine M_m started with empty tape enumerates all proofs of S and records all proven statements p. If it finds proofs of p and $\sim p$ for some p it halts. Then

$$consis(S) :\equiv H(m\#)$$

□

Lemma 8. *If machine M_u started with v halts, then one can prove this in Z_e.*

$$H(u\#v) \ true \Rightarrow S \vdash H(u\#v)$$

Proof. One proves this by writing down a sequence of configurations and checking that it is the computation on M_u started with v. This argument can be formalized in Z_E. □

3 Towards the second incompleteness theorem

3.1 Preliminaries

For proof systems S denote by $consis(S)$ the statement, that S is consistent.

Lemma 7. *Consistency can be formulated in Z_E*

$$consis(S) \in L_E$$

Proof. Machine M_m started with empty tape enumerates all proofs of S and records all proven statements p. If it finds proofs of p and $\sim p$ for some p it halts. Then

$$consis(S) :\equiv H(m\#)$$

□

Lemma 8. *If machine M_u started with v halts, then one can prove this in Z_e.*

$$H(u\#v) \; true \Rightarrow S \vdash H(u\#v)$$

Proof. One proves this by writing down a sequence of configurations and checking that it is the computation on M_u started with v. This argument can be formalized in Z_E. □

Lemma 9. *If p is provable in Z_E, then one can prove, that it is provable*

$$S \vdash p \Rightarrow (S \vdash (S \vdash p))$$

Proof. One establishes the existence of a proof by writing down the proof and then checking (with a syntax check), that the proof rules are obeyed. This argument can be formalized in Z_e □

3.2 An explicit statement, which is independent of Z_E

def: independence Let $S = (\Sigma, L, A, R)$ be a proof system. A statement $v \in L$ is called independent of S if it can be neither proved nor disproved in S.

$$\sim S \vdash v \wedge \sim S \vdash \sim v$$

S is incomplete iff independent p exists. From now on assume that S extends Z_E, i.e. it is at least as powerful as Z_E.

$$Z_E \subseteq S$$

3.2 An explicit statement, which is independent of Z_E

def: independence Let $S = (\Sigma, L, A, R)$ be a proof system. A statement $v \in L$ is called independent of S if it can be neither proved nor disproved in S.

$$\sim S \vdash v \wedge \sim S \vdash \sim v$$

S is incomplete iff independent p exists. From now on assume that S extends Z_E, i.e. it is at least as powerful as Z_E.

$$Z_E \subseteq S$$

a statement: Consider Turing machine $Q = M_q$. Started with w it enumerates all provable statements of S, i.e. the set

$$P = \{v \,:\, S \vdash v\}$$

It halts as soon as it finds a proof of $\sim H(w\#w)$.

Lemma 10. *If S is correct, then statement $H(q\#q)$ is independent of S.*

Q started with its own program q halts
as soon as it finds a proof
that it does not halt

3.2 An explicit statement, which is independent of Z_E

def: independence Let $S = (\Sigma, L, A, R)$ be a proof system. A statement $v \in L$ is called independent of S if it can be neither proved nor disproved in S.

$$\sim S \vdash v \wedge \sim S \vdash \sim v$$

S is incomplete iff independent p exists. From now on assume that S extends Z_E, i.e. it is at least as powerful as Z_E.

$$Z_E \subseteq S$$

a statement: Consider Turing machine $Q = M_q$. Started with w it enumerates all provable statements of S, i.e. the set

$$P = \{v \ : \ S \vdash v\}$$

It halts as soon as it finds a proof of $\sim H(w\#w)$.

Lemma 10. *If S is correct, then statement $H(q\#q)$ is independent of S.*

Q started with its own program q halts
as soon as it finds a proof
that it does not halt

- assume $S \vdash H(q\#q)$ then

$$\Rightarrow \ \sim S \vdash \sim H(q\#q) \quad (S \text{ correct})$$
$$\Rightarrow \ \sim H(q\#q) \quad (\text{construction of } Q)$$
$$\Rightarrow \ S \text{ not correct}$$

3.2 An explicit statement, which is independent of Z_E

def: independence Let $S = (\Sigma, L, A, R)$ be a proof system. A statement $v \in L$ is called independent of S if it can be neither proved nor disproved in S.

$$\sim S \vdash v \wedge \sim S \vdash \sim v$$

S is incomplete iff independent p exists. From now on assume that S extends Z_E, i.e. it is at least as powerful as Z_E.

$$Z_E \subseteq S$$

a statement: Consider Turing machine $Q = M_q$. Started with w it enumerates all provable statements of S, i.e. the set

$$P = \{v \,:\, S \vdash v\}$$

It halts as soon as it finds a proof of $\sim H(w\#w)$.

Lemma 10. *If S is correct, then statement $H(q\#q)$ is independent of S.*

- assume $S \vdash H(q\#q)$ then

$$\begin{aligned} &\Rightarrow \quad \sim S \vdash \sim H(q\#q) \quad (S \text{ correct}) \\ &\Rightarrow \quad \sim H(q\#q) \quad (\text{construction of } Q) \\ &\Rightarrow \quad S \text{ not correct} \end{aligned}$$

- assume $S \vdash \sim H(q\#q)$ then

$$\begin{aligned} &\Rightarrow \quad Q \text{ finds proof} \\ &\Rightarrow \quad H(q\#q) \quad (\text{construction of } Q) \\ &\Rightarrow \quad S \text{ not correct} \end{aligned}$$

Q started with its own program q halts
as soon as it finds a proof
that it does not halt

3.2 An explicit statement, which is independent of Z_E

def: independence Let $S = (\Sigma, L, A, R)$ be a proof system. A statement $v \in L$ is called independent of S if it can be neither proved nor disproved in S.

$$\sim S \vdash v \wedge \sim S \vdash \sim v$$

S is incomplete iff independent p exists. From now on assume that S extends Z_E, i.e. it is at least as powerful as Z_E.

$$Z_E \subseteq S$$

a statement: Consider Turing machine $Q = M_q$. Started with w it enumerates all provable statements of S, i.e. the set

$$P = \{v \ : \ S \vdash v\}$$

It halts as soon as it finds a proof of $\sim H(w\#w)$.

Lemma 10. *If S is correct, then statement $H(q\#q)$ is independent of S.*

- assume $S \vdash H(q\#q)$ then

$$\Rightarrow \ \sim S \vdash \sim H(q\#q) \quad (S \text{ correct})$$
$$\Rightarrow \ \sim H(q\#q) \quad (\text{construction of } Q)$$
$$\Rightarrow \ S \text{ not correct}$$

- assume $S \vdash \sim H(q\#q)$ then

$$\Rightarrow \ Q \text{ finds proof}$$
$$\Rightarrow \ H(q\#q) \quad (\text{construction of } Q)$$
$$\Rightarrow \ S \text{ not correct}$$

Lemma 11. *Statement $H(q\#q)$ is false*

- assume M_q started with q halts,
- lemma 8: $H(q\#q)$ is provable
- impossible by lemma 10

a statement: Consider Turing machine $Q = M_q$. Started with w it enumerates all provable statements of S, i.e. the set

$$P = \{v \;:\; S \vdash v\}$$

It halts as soon as it finds a proof of $\sim H(w\#w)$.

Lemma 10. *If S is correct, then statement $H(q\#q)$ is independent of S.*

3.3 weakening correctness to consistency

Lemma 12. *If S is consistent, then statement $\sim H(q\#q)$ is not provable in S.*

$$consis(S) \Rightarrow \sim S \vdash \sim H(q\#q)$$

a statement: Consider Turing machine $Q = M_q$. Started with w it enumerates all provable statements of S, i.e. the set

$$P = \{v \;:\; S \vdash v\}$$

It halts as soon as it finds a proof of $\sim H(w\#w)$.

Lemma 10. *If S is correct, then statement $H(q\#q)$ is independent of S.*

3.3 weakening correctness to consistency

Lemma 12. *If S is consistent, then statement $\sim H(q\#q)$ is not provable in S.*

$$consis(S) \Rightarrow \sim S \vdash \sim H(q\#q)$$

asssume $S \vdash \sim H(q\#q)$. Then

$$\begin{aligned} &\Rightarrow\quad Q \text{ finds proof and halts} \\ &\Rightarrow\quad H(q\#q) \\ &\Rightarrow\quad S \vdash H(q\#q) \quad \text{(lemma 8)} \\ &\Rightarrow\quad \sim consis(S) \end{aligned}$$

a statement: Consider Turing machine $Q = M_q$. Started with w it enumerates all provable statements of S, i.e. the set

$$P = \{v \, : \, S \vdash v\}$$

It halts as soon as it finds a proof of $\sim H(w\#w)$.

Lemma 10. *If S is correct, then statement $H(q\#q)$ is independent of S.*

3.3 weakening correctness to consistency

Lemma 12. *If S is consistent, then statement $\sim H(q\#q)$ is not provable in S.*

$$consis(S) \Rightarrow \sim S \vdash \sim H(q\#q)$$

asssume $S \vdash \sim H(q\#q)$. Then

$$\begin{aligned} &\Rightarrow \quad Q \text{ finds proof and halts} \\ &\Rightarrow \quad H(q\#q) \\ &\Rightarrow \quad S \vdash H(q\#q) \quad \text{(lemma 8)} \\ &\Rightarrow \quad \sim consis(S) \end{aligned}$$

Lemma 13. *Lemma 12 is provable in S*

$$S \vdash (consis(S) \rightarrow \sim S \vdash \sim H(q\#q))$$

Proof. The proof of lemma 12 can be formalized in Z_E. □

a statement: Consider Turing machine $Q = M_q$. Started with w it enumerates all provable statements of S, i.e. the set

$$P = \{v \,:\, S \vdash v\}$$

It halts as soon as it finds a proof of $\sim H(w\#w)$.

Lemma 10. *If S is correct, then statement $H(q\#q)$ is independent of S.*

3.3 weakening correctness to consistency

Lemma 13. *Lemma 12 is provable in S*

$$S \vdash (consis(S) \rightarrow \sim S \vdash \sim H(q\#q))$$

3.4 The second incompleteness theorem

Lemma 14. *If $Z_E \subseteq S$ and S is consistent, then one cannot prove the consistencyy of S in S*

$$consis(S) \rightarrow \sim S \vdash consis(S)$$

a statement: Consider Turing machine $Q = M_q$. Started with w it enumerates all provable statements of S, i.e. the set

$$P = \{v \; : \; S \vdash v\}$$

It halts as soon as it finds a proof of $\sim H(w\#w)$.

Lemma 10. *If S is correct, then statement $H(q\#q)$ is independent of S.*

3.3 weakening correctness to consistency

Lemma 13. *Lemma 12 is provable in S*

$$S \vdash (consis(S) \rightarrow \sim S \vdash \sim H(q\#q))$$

3.4 The second incompleteness theorem

Lemma 14. *If $Z_E \subseteq S$ and S is consistent, then one cannot prove the consistencyy of S in S*

$$consis(S) \rightarrow \sim S \vdash consis(S)$$

- assume

$$S \vdash \underbrace{consis(S)}_{A}$$

- lemma 13:

$$S \vdash (consis(S) \rightarrow \underbrace{\sim S \vdash \sim H(q\#q)}_{B})$$

a statement: Consider Turing machine $Q = M_q$. Started with w it enumerates all provable statements of S, i.e. the set

$$P = \{v \,:\, S \vdash v\}$$

It halts as soon as it finds a proof of $\sim H(w\#w)$.

Lemma 10. *If S is correct, then statement $H(q\#q)$ is independent of S.*

3.3 weakening correctness to consistency

Lemma 13. *Lemma 12 is provable in S*

$$S \vdash (consis(S) \rightarrow \sim S \vdash \sim H(q\#q))$$

3.4 The second incompleteness theorem

Lemma 14. *If $Z_E \subseteq S$ and S is consistent, then one cannot prove the consistencyy of S in S*

$$consis(S) \rightarrow \sim S \vdash consis(S)$$

- assume

$$S \vdash \underbrace{consis(S)}_{A}$$

- lemma 13:

$$S \vdash (consis(S) \rightarrow \underbrace{\sim S \vdash \sim H(q\#q)}_{B})$$

- modus ponens

$$\frac{A\ , A \rightarrow B}{B}$$

-

$$S \vdash (\underbrace{\sim S \vdash \sim H(q\#q)}_{B})$$

- construction of Q:

$$B \rightarrow \underbrace{\sim H(q\#q)}_{C}$$

a statement: Consider Turing machine $Q = M_q$. Started with w it enumerates all provable statements of S, i.e. the set

$$P = \{v \ : \ S \vdash v\}$$

It halts as soon as it finds a proof of $\sim H(w\#w)$.

Lemma 10. *If S is correct, then statement $H(q\#q)$ is independent of S.*

3.3 weakening correctness to consistency

Lemma 13. *Lemma 12 is provable in S*

$$S \vdash (consis(S) \to \sim S \vdash \sim H(q\#q))$$

3.4 The second incompleteness theorem

Lemma 14. *If $Z_E \subseteq S$ and S is consistent, then one cannot prove the consistencyy of S in S*

$$consis(S) \to \sim S \vdash consis(S)$$

- assume

$$S \vdash \underbrace{consis(S)}_{A}$$

- lemma 13:

$$S \vdash (consis(S) \to \underbrace{\sim S \vdash \sim H(q\#q)}_{B})$$

- modus ponens

$$\frac{A \ , A \to B}{B}$$

-

$$S \vdash (\underbrace{\sim S \vdash \sim H(q\#q)}_{B})$$

- construction of Q:

$$B \to \underbrace{\sim H(q\#q)}_{C}$$

- this argument can be formalized in Z_E

$$S \vdash B \to C$$

a statement: Consider Turing machine $Q = M_q$. Started with w it enumerates all provable statements of S, i.e. the set

$$P = \{v \,:\, S \vdash v\}$$

It halts as soon as it finds a proof of $\sim H(w\#w)$.

Lemma 10. *If S is correct, then statement $H(q\#q)$ is independent of S.*

3.3 weakening correctness to consistency

Lemma 13. *Lemma 12 is provable in S*

$$S \vdash (consis(S) \rightarrow \sim S \vdash \sim H(q\#q))$$

3.4 The second incompleteness theorem

Lemma 14. *If $Z_E \subseteq S$ and S is consistent, then one cannot prove the consistencyy of S in S*

$$consis(S) \rightarrow \sim S \vdash consis(S)$$

- assume

$$S \vdash \underbrace{consis(S)}_{A}$$

- lemma 13:

$$S \vdash (consis(S) \rightarrow \underbrace{\sim S \vdash \sim H(q\#q))}_{B}$$

- modus ponens

$$\frac{A \,, A \rightarrow B}{B}$$

-

$$S \vdash \underbrace{(\sim S \vdash \sim H(q\#q))}_{B}$$

- construction of Q:

$$B \rightarrow \underbrace{\sim H(q\#q)}_{C}$$

- this argument can be formalized in Z_E

$$S \vdash B \rightarrow C$$

- modus ponens

$$\frac{B \,, B \rightarrow C}{C}$$

-

$$S \vdash \sim H(q\#q)$$

- lemma 9 (provability is provable)

$$S \vdash \underbrace{(S \vdash \sim H(q\#q))}_{\sim B}$$

S not consistent

warning

before you try to find such proofs yourself
for extended periods of time:

read vita of Cantor and Gödel in Wikipedia

there might be a mental health hazard

4 A non standard natural number

theories (as considered here) have

- a universe U. Here natural numbers.
- functions f with arguments and values in U, Here $+, \times$
- theory is *countable* if the set of axioms and proof rules in countable

4 A non standard natural number

theories (as considered here) have

- a universe U. Here natural numbers.
- functions f with arguments and values in U, Here $+, \times$
- theory is *countable* if the set of axioms and proof rules in countable

model of a theory : instantiation of universe U and functions f such that the axioms hold. For Z_E

- decimal numbers
- binary numbers
- unary numbers: finite sequences of vertical lines |
- all *isomorphic* elements of U in different models are just renamings of each other
- standard natural numbers

4 A non standard natural number

theories (as considered here) have

- a universe U. Here natural numbers.
- functions f with arguments and values in U, Here $+, \times$
- theory is *countable* if the set of axioms and proof rules in countable

model of a theory : instantiation of universe U and functions f such that the axioms hold. For Z_E

- decimal numbers
- binary numbers
- unary numbers: finite sequences of vertical lines |
- all *isomorphic* elements of U in different models are just renamings of each other
- standard natural numbers

do non standard models of Z_E exist?

4 A non standard natural number

theories (as considered here) have

- a universe U. Here natural numbers.
- functions f with arguments and values in U, Here $+, \times$
- theory is *countable* if the set of axioms and proof rules in countable

model of a theory : instantiation of universe U and functions f such that the axioms hold. For Z_E

- decimal numbers
- binary numbers
- unary numbers: finite sequences of vertical lines |
- all *isomorphic* elements of U in different models are just renamings of each other
- standard natural numbers

do non standard models of Z_E exist?

model existence theorem of logic:

Lemma 15. *Every consistent countable theory has a countable model.*

4 A non standard natural number

theories (as considered here) have

- a universe U. Here natural numbers.
- functions f with arguments and values in U, Here $+, \times$
- theory is *countable* if the set of axioms and proof rules in countable

model of a theory : instantiation of universe U and functions f such that the axioms hold. For Z_E

- decimal numbers
- binary numbers
- unary numbers: finite sequences of vertical lines |
- all *isomorphic* elements of U in different models are just renamings of each other
- standard natural numbers

do non standard models of Z_E exist?

model existence theorem of logic:

Lemma 15. *Every consistent countable theory has a countable model.*

- define $\boxed{\alpha}$ as the number of steps after which Q started with q halts.
- does not exist, because $H(q\#q)$ is false
- the statement that it exists is consistent.
- adding this statement to axioms of Z_E gives a consistent theory.
- has model by model existence theorem.
- it contains this 'step number'

5 A countable model of set theory and the real numbers

model existence theorem of logic:

Lemma 15. *Every consistent countable theory has a countable model.*

5 A countable model of set theory and the real numbers

model existence theorem of logic:

Lemma 15. *Every consistent countable theory has a countable model.*

- established proof systems for set teory (including the reals) exist
- e.g. ZF: Zermelo-Fraenkel set theory
- countable!
- so there exist countable models

5 A countable model of set theory and the real numbers

model existence theorem of logic:

Lemma 15. *Every consistent countable theory has a countable model.*

- established proof systems for set teory (including the reals) exist
- e.g. ZF: Zermelo-Fraenkel set theory
- countable!
- so there exist countable models

how can this be?

5 A countable model of set theory and the real numbers

model existence theorem of logic:

Lemma 15. *Every consistent countable theory has a countable model.*

- established proof systems for set teory (including the reals) exist
- e.g. ZF: Zermelo-Fraenkel set theory
- countable!
- so there exist countable models

how can this be?

- in a langue L you can only define countably many entities
- include in model only the ones you can define/talk about

5 A countable model of set theory and the real numbers

model existence theorem of logic:

Lemma 15. *Every consistent countable theory has a countable model.*

- established proof systems for set teory (including the reals) exist
- e.g. ZF: Zermelo-Fraenkel set theory
- countable!
- so there exist countable models

how can this be?

- in a langue L you can only define countably many entities
- include in model only the ones you can define/talk about

- question: do things you cannot talk about, really exist?