Recursion Theory

Joost J. Joosten

Institute for Logic Language and Computation

University of Amsterdam

Plantage Muidergracht 24

1018 TV Amsterdam

Room P 3.26, +31 20 5256095

jjoosten@phil.uu.nl

www.phil.uu.nl/∼jjoosten

Questions

- Any questions?

Questions

- Any questions?
- Midterm exam next week

Questions

- Any questions?
- Midterm exam next week
- Different time slot

Questions

- Any questions?
- Midterm exam next week
- Different time slot
- tuesday October 24, 13.30-16.30, OMHP-C017 (OudeManHuisPoort 4-6)

Questions

- Any questions?
- Midterm exam next week
- Different time slot
- tuesday October 24, 13.30-16.30, OMHP-C017 (OudeManHuisPoort 4-6)
- Workgroup as normal?

Questions

- Any questions?
- Midterm exam next week
- Different time slot
- tuesday October 24, 13.30-16.30, OMHP-C017 (OudeManHuisPoort 4-6)
- Workgroup as normal?
- Or does someone have an exam at that time?

Questions

- Any questions?
- Midterm exam next week
- Different time slot
- tuesday October 24, 13.30-16.30, OMHP-C017 (OudeManHuisPoort 4-6)
- Workgroup as normal?
- Or does someone have an exam at that time?
- I will publish an exercise mid-term exam shortly on my webpage

Characterizing c.e. sets

- We introduce some notation to characterize the c.e. sets

Characterizing c.e. sets

- We introduce some notation to characterize the c.e. sets
- First: W_e, the halting set of e

Characterizing c.e. sets

- We introduce some notation to characterize the c.e. sets
- First: W_e, the halting set of e
- Next: a relation being Σ^0_1, Π^0_1 or Δ^0_1

Characterizing c.e. sets

- We introduce some notation to characterize the c.e. sets
- First: W_e, the halting set of e
- Next: a relation being Σ^0_1, Π^0_1 or Δ^0_1
- Example: for a given e, the set $\{x \mid \varphi_e(x) \downarrow\}$ is Σ_1

Characterizing c.e. sets

- We introduce some notation to characterize the c.e. sets
- First: W_e, the halting set of e
- Next: a relation being Σ_1^0, Π_1^0 or Δ_1^0
- Example: for a given e, the set $\{x \mid \varphi_e(x)\downarrow\}$ is Σ_1
- Proof: $\varphi_e(x)\downarrow$ iff $(\exists s)\,(\exists y)\;\; \varphi_{e,s}(x) = y$

Normal Form Theorem

- The NFT states the equivalence of the following three statements for any set A

Normal Form Theorem

- The NFT states the equivalence of the following three statements for any set A
- (1) A is c.e.

Normal Form Theorem

- The NFT states the equivalence of the following three statements for any set A
- (1) A is c.e.
- (2) A is Σ_1^0

Normal Form Theorem

- The NFT states the equivalence of the following three statements for any set A
- (1) A is c.e.
- (2) A is Σ_1^0
- (3) A is W_e for some e

Normal Form Theorem

- The NFT states the equivalence of the following three statements for any set A
- (1) A is c.e.
- (2) A is Σ_1^0
- (3) A is W_e for some e
- Proof:

Normal Form Theorem

- The NFT states the equivalence of the following three statements for any set A
- (1) A is c.e.
- (2) A is Σ^0_1
- (3) A is W_e for some e
- Proof:
- (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (1)

A Universal c.e. set

- Theorem (Enumeration Theorem for c.e. sets)

A Universal c.e. set

- Theorem (Enumeration Theorem for c.e. sets)
- There exists a c.e. set K_0 such that

$$W_e = \{x \mid \langle e, x \rangle \in K_0\}$$

A Universal c.e. set

- Theorem (Enumeration Theorem for c.e. sets)
- There exists a c.e. set K_0 such that

$$W_e = \{x \mid \langle e, x \rangle \in K_0\}$$

- Proof: Define K_0 as expected and use the NFT to show that it is c.e.

Incomputable sets

- **Theorem**: K is an incomputable c.e. set

Incomputable sets

- **Theorem**: K is an incomputable c.e. set
- Proof:

Incomputable sets

- **Theorem**: K is an incomputable c.e. set
- Proof:
- It is clear that K is c.e.

Incomputable sets

- **Theorem**: K is an incomputable c.e. set
- Proof:
- It is clear that K is c.e.
- However, its complement $\overline{K}$ is not.

Incomputable sets

- **Theorem**: K is an incomputable c.e. set
- Proof:
- It is clear that K is c.e.
- However, its complement $\overline{K}$ is not.
- This is again a diagonal argument

Incomputable sets

- **Theorem**: K is an incomputable c.e. set
- Proof:
- It is clear that K is c.e.
- However, its complement $\overline{K}$ is not.
- This is again a diagonal argument

- **Corollary**: There is a Turing machine with an unsolvable halting problem

Incomputable sets

- **Theorem**: K is an incomputable c.e. set
- Proof:
- It is clear that K is c.e.
- However, its complement $\overline{K}$ is not.
- This is again a diagonal argument

- **Corollary**: There is a Turing machine with an unsolvable halting problem
- **Corollary**: The halting problem for the universal TM is unsolvable

A ‘natural’ incomputable function

- Busy beaver function

A ‘natural’ incomputable function

- Busy beaver function
- B(n) is the maximal output on input 0 of an URM whose program contains at most n lines

A ‘natural’ incomputable function

- Busy beaver function
- B(n) is the maximal output on input 0 of an URM whose program contains at most n lines
- Well defined?

A ‘natural’ incomputable function

- Busy beaver function
- B(n) is the maximal output on input 0 of an URM whose program contains at most n lines
- Well defined?
- Lemma: for each n there are only finitely many functions described by programs containing no more than n lines.

A ‘natural’ incomputable function

- Busy beaver function
- B(n) is the maximal output on input 0 of an URM whose program contains at most n lines
- Well defined?
- Lemma: for each n there are only finitely many functions described by programs containing no more than n lines.
- B dominates all URM computable functions

A ‘natural’ incomputable function

- Busy beaver function
- B(n) is the maximal output on input 0 of an URM whose program contains at most n lines
- Well defined?
- Lemma: for each n there are only finitely many functions described by programs containing no more than n lines.
- B dominates all URM computable functions
- Can someone show $B(10) \geq 39$?

A ‘natural’ incomputable function

- Busy beaver function
- B(n) is the maximal output on input 0 of an URM whose program contains at most n lines
- Well defined?
- Lemma: for each n there are only finitely many functions described by programs containing no more than n lines.
- B dominates all URM computable functions
- Can someone show $B(10) \geq 39$?
- Note: B is strictly increasing

A ‘natural’ incomputable function

- Busy beaver function
- B(n) is the maximal output on input 0 of an URM whose program contains at most n lines
- Well defined?
- Lemma: for each n there are only finitely many functions described by programs containing no more than n lines.
- B dominates all URM computable functions
- Can someone show $B(10) \geq 39$?
- Note: B is strictly increasing
- Lemma: $B(n+5) \geq 2 \cdot n$

Busy Beaver

- Lemma: every URM computable function is dominated by some strictly increasing URM computable function

Busy Beaver

- Lemma: every URM computable function is dominated by some strictly increasing URM computable function
- Tibor Radó (1962): The BB function dominates any URM computable function

Busy Beaver

- Lemma: every URM computable function is dominated by some strictly increasing URM computable function
- Tibor Radó (1962): The BB function dominates any URM computable function
- Proof:

Busy Beaver

- Lemma: every URM computable function is dominated by some strictly increasing URM computable function
- Tibor Radó (1962): The BB function dominates any URM computable function
- Proof: Suppose P_g has k_0 lines
- Then $B(n+k_0) \geq g(B(n))$

Busy Beaver

- Lemma: every URM computable function is dominated by some strictly increasing URM computable function
- Tibor Radó (1962): The BB function dominates any URM computable function
- Proof: Suppose P_g has k_0 lines
- Then $B(n+k_0) \geq g(B(n))$
- composing with other facts yields the answer