

Advanced Topics in Theoretical Computer Science

Part 4: Computability and (Un-)Decidability

13.12.2017

Viorica Sofronie-Stokkermans

Universität Koblenz-Landau

e-mail: sofronie@uni-koblenz.de

Last time

- Recall: Turing machines and Turing computability
- Register machines (LOOP, WHILE, GOTO)
- **Recursive functions**
- The Church-Turing Thesis
- Computability and (Un-)decidability
- Complexity
- Other computation models: e.g. Büchi Automata

Last time

- Recall: Turing machines and Turing computability
- Register machines (LOOP, WHILE, GOTO)
- Recursive functions
- **The Church-Turing Thesis**
- Computability and (Un-)decidability
- Complexity
- Other computation models: e.g. Büchi Automata

Today

- Recapitulation: Turing machines and Turing computability
- Recursive functions
- Register machines (LOOP, WHILE, GOTO)
- The Church-Turing Thesis
- **Computability and (Un-)decidability**
- Complexity
- Other computation models: e.g. Büchi Automata

Computability and (Un-)decidability

Known undecidable problems (Theoretical Computer Science I)

- The halting problem for Turing machines
- Equivalence problem

Computability and (Un-)decidability

Known undecidable problems (Theoretical Computer Science I)

- The halting problem for Turing machines
- The equivalence problem

Consequences:

- All problems about programs (TM) which are non-trivial (in a certain sense) are undecidable (Theorem of Rice)
- Identify undecidable problems outside the world of Turing machines
 - Validity/Satisfiability in First-Order Logic
 - The Post Correspondence Problem

Computability and (Un-)decidability

Known undecidable problems (Theoretical Computer Science I)

- The halting problem for Turing machines
- The equivalence problem

Consequences:

- All problems about programs (TM) which are non-trivial (in a certain sense) are undecidable (Theorem of Rice)
- Identify undecidable problems outside the world of Turing machines
 - Validity/Satisfiability in First-Order Logic
 - The Post Correspondence Problem

Computability and (Un-)decidability

The Theorem of Rice (informally)

Variant 1

For each non-trivial property P of languages of type 0:

It is undecidable, whether the language accepted by a Turing machine has property P .

This variant will be formalized and proved in this lecture.

Computability and (Un-)decidability

The Theorem of Rice (informally)

Variant 2

For each non-trivial property P of (partial) functions:

It is undecidable, whether the function computed by a Turing machine has property P .

Generalization of Variant 2:

The same holds for other computability models:

- algorithms
- Java programs
- λ expressions
- recursive functions
- etc.

Computability and (Un-)decidability

The Theorem of Rice (informally)

Variant 2

For each non-trivial property P of (partial) functions:

It is undecidable, whether the function computed by a Turing machine has property P .

Generalization of Variant 2:

The same holds for other computability models:

- algorithms
- Java programs
- λ expressions
- recursive functions
- etc.

Examples of non-trivial properties of functions:

- Monotonicity: $\{f \mid f(i) \leq f(i+1) \text{ for all } i\}$
- Equivalence: $\{f \mid f = g\}$ for a given g
- Image: $\{f \mid j \in \text{Im}(f)\}$ for a given j
- Square function $\{f \mid f(i) = i^2 \text{ for all } i\}$

Plan

- Recall: Acceptance and Decidability
- Recall: Undecidability results
 - The halting problem
 - Undecidability proofs via reduction
- The theorem of Rice (Variant 1)