Before Starting

- Questions
- Computational problems as sets
  - Need to know the answer in advance?
  - Representing the relation between inputs and the problem schematically?

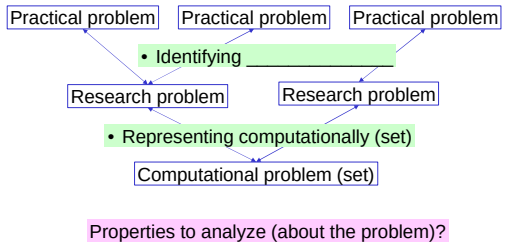
Unit A3: Overview

- Discuss your ideas about “Theory of Computation”
- Preview the main components of the Theory of Computation
- Practice representing computational problems as sets
- Preview Exercise A3 “Computational problem solving”

Exercise A1 Part 1

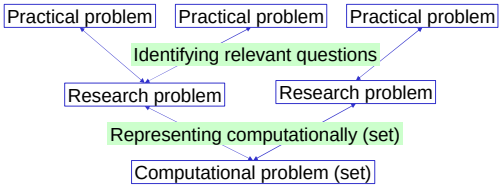
- Your ideas about Theory of Computation
  - Task 1 (review): Identify/analyze CS “theories” you know
  - Task 2 (~ Group Exercise 3): Usefulness of the theories you know; missing elements

Problems and Theory



Properties to analyze (about the problem)?

Problems and Theory

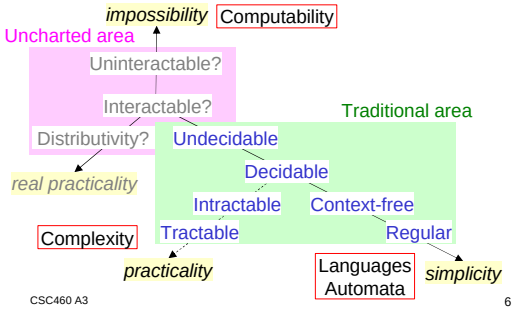


- Possible to compute at all?
- How to choose the simplest mechanism?
- Practical for any future inputs?

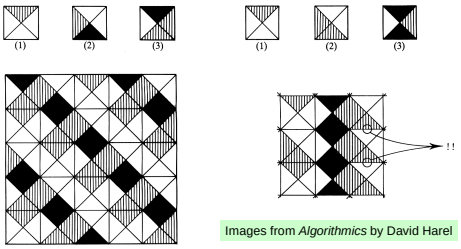
Theory

Significance?

Preview: Theory of Computation



Example 1: Floor Tiling



Images from *Algorithmics* by David Harel

Cf. some of your own problems

Computability

- Content Goal 4
- Example questions
  - What are the limitations of computation? All the computational problems (sets) solvable?
  - How can we compare different forms of "computation?"
  - What is the notion of "computation?"
- Significance?
- Are your own problems "computable?"

Schematic representation

Example 2: Survivor

- Scenario
  - Alone on a large island where you survived an aircraft crash.
  - Established a base where you can spend your nights safely.
  - Still need to explore the island to obtain foods.
- Appropriate (minimal) mechanism for your "mental" computer?

Cf. some of your own problems

Languages/Automata

- Content Goal 5
- Example questions
  - Where are the balance between the simplicity of computational mechanisms and their abilities?
  - How can we identify an appropriate computational mechanism for a given problem (set).
- Significance?
- The simplest mechanism for your own problems?

Example 3: Scheduling

- How to define the problem?
- Must be sufficiently general to be able to handle cases including:
  - Your time management
  - Organizing a team of workers
  - Assigning courses to instructors
  - Schedule the analysis process for the entire human genome

Cf. some of your own problems

Complexity

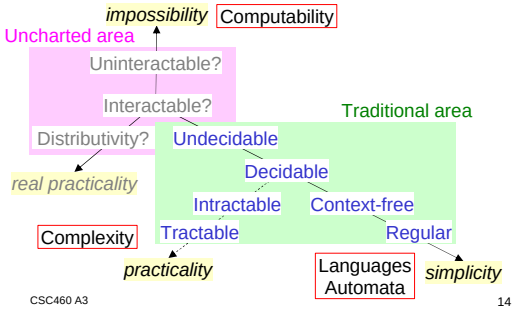
- Content Goal 6
- Example questions
  - How does the input data size affect the computation?
  - What would be the limit of "practical" computation?
- Significance?
- Practicality of your own problems, esp. facing large data?

What is not usually discussed...

- Content Goal 3: Interactivity
- Example questions
  - How well does algorithmic computation fare in real-world computation?
  - What are the essential properties that are needed to solve real-world problems computationally?
- Significance? Examples? Implications?

These questions as computational problems?

Preview: Theory of Computation



Interim Summary

- Theory of Computation
  - Offers principles in three main aspects: computability, languages/automata, complexity, most commonly dealing with problems as sets
  - Limitations due to its traditional foundation on algorithmic computation, cf. interactivity
- Caveat
  - Focus on computational problems (sets), i.e., no systematic methods for problem transformation, cf. discrete math

Practice: Robotics

- Suppose that robotic control of an automobile has been transformed into a computational problem (set).
- Analyze with respect to the following:
  - Computability (possibility)
  - Languages/Automata (simplest mechanism)
  - Complexity (practicality with respect to the input size)

Group Exercise 1

- Suppose that a typical compilation problem has been transformed into a computational problem (set).
- Analyze with respect to the following:
  - Computability (possibility)
  - Languages/Automata (simplest mechanism)
  - Complexity (practicality with respect to the input size)

Set Representation of Problems

- Benefits
  - To analyze/compute such problems, we only need a single mechanism to check set membership
  - Well-developed techniques in mathematics are available
- Limitations?

Alternative Forms

- Computational problem
- Set
- Language
  - Set of strings (special case of set)
- Characteristic function
  - Outputs yes/no, i.e., comparable to a relation (set of tuples)

Set Examples

1. What is the sex of Furby?
   {*female*}
2. What are palindromes?
   {*a*, *cc*, *wow*, *abba*, *kayak*, *hannah*, ...}
3. Addition operation (on natural numbers)
   {(0, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 2), ...}
4. Causal relation (on any possible event)
   {(*drink*, *joy*), (*drink*, *nausea*), (*drink*, *addiction*), ...}

Set Notation (Review)

- List notation (technically, only for finite sets)
  - {1, 2, 3, 4, 5}
  - {(paper, stone), (scissors, paper), (stone, scissors)}
- Predicate notation
  - $\{x \mid 0 < x \leq 5, x \text{ is a natural number}\}$
  - $\{(x, y) \mid x \text{ wins over } y \text{ in the paper-scissors-stone game}\}$
  - $\{x \mid x \text{ is a number never used by the human}\}$

Predicate notation as a filter

Practice: Predicate Notation

- Palindromes
  {a, cc, wow, abba, kayak, hannah, ...}
  = {x | x is a palindrome} A better way?
  = { }
- Addition operation (on natural numbers)
  {(0, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 2), ...}
  = { }
- Causal relation (on any possible event)
  {(drink, joy), (drink, nausea), (drink, addiction), ...}
  = { }

Group Exercise 2

- Infinite loop is one of the most common bugs in any program. It would be enormously helpful if someone writes a program to detect infinite loops in a given program. **Give** the set representation of the computational problem involved here. **Speculate** the basic Theory properties (i.e., possibility, etc.), referring to the set (a concise, informal description suffices).

Group Exercise 3

- For any (possibly very large) academic department, how can we assign all the courses to the instructors within the usual constraints (e.g., a single instructor for a single course)? **Give** the set representation of the computational problem involved here. **Speculate** the basic Theory properties, referring to the set.

Unit Summary

- Theory of Computation: computablity, languages/automata, complexity
- Set representation of computational problems: list/predicate notations
- Understand Unit A3 Exercise
- Summary question
  - We discussed broad ideas with little details. So, you must have questions or be uncertain on at least some aspect. What are they?