What to do with your problems?

Unit A1: Overview

- Discuss your own problems
- Analyze the process of going from problems to solutions
- Identify different types of problems
- Discuss how to represent problems
- Practice transforming problems
- Preview Exercise A1 "Using a Theory"

Questions about the course? (Ex00 Part 2)

Exercise 00 Part 1

- Your own problems
- Analysis of computational aspects (optional)
  - Can obtain an answer?
  - Can do it with available resources?
  - Can do it within a reasonable time?
  - Can handle different cases of the problem?

Why problems?

Problems

- Problem transformation
  - From more realistic to more manageable
- Problem types (e.g., use of computers)
  - Practical problems: Calls for an *action* (in reality)
  - Research problems: Calls for *information*
  - Computational problems: Calls for *computation*

Research Problems

- As questions
  - *Yes-no* question: "Is Furby male?"
  - *Wh*-question: "What is the sex of Furby?"
- Cost *or* significance
  - Suffering of not being able to answer *or* benefit of answering [opposite sides]
  - Often, in connection to a practical problem
- Example: "How do ants find food?"

"The Craft of Research" (see the topic list)

Digression: Analogy

- Problem (question) and significance
- Course (materials) and significance
- Organization and role
- Anatomy and physiology

Feeding problems to a computer

Computational Problems

Preliminary

- Research problem 1: "Is Furby male?"
  - Computational problem (à la Prolog)
    - Input: **`furbySex(male).`**
    - Output: e.g., **`no.`**

Suppose the computer knows the answer

- Research problem 2: "What is the sex of Furby?"
  - Computational problem (à la Prolog)
    - Input: **`furbySex(X).`**
    - Output: e.g., **`X = female.`**

Imagine a Lazy Machine

- Only responds with "uh" or "nah"
- Question: Can we phrase various problems in a way even the lazy machine can still be useful? If so, how?

Possibility of dealing with problems in a more uniform manner

Converting Questions

- Any question (including *wh*-questions) can be converted into a series of *yes-no* questions.
  - *Wh*-question: "What is the sex of Furby?"
  - *Yes-no* questions: "Is Furby male?" "Is Furby female?"
- This technique allows us to deal with complex problems with "lazy" machines.

Group Exercise 1

- Convert the following *wh*-questions into a series of *yes-no* questions
  1. What are the planets (of the Sun)?
  2. Where did you eat?
  3. When will the last human be born?

Representing formally?

Formal Representation

- *wh*: "What is the square of x?" — Function
  - $f = \{(1, 1), (2, 4), (3, 9), ...\}$
  - I.e., $f(1) = 1$, $f(2) = 4$, $f(3) = 9$, ...
  - Pairs of an input and the output
- *yes-no*: "Is y the square of x?" — Relation
  - $R = \{(1, 1), (2, 4), (3, 9), ...\}$
  - I.e., $(1, 1) \in R$, $(2, 4) \in R$, $(3, 9) \in R$, ...
  - Pairs of two inputs

Discrete math: review if necessary

Number of Arguments

- 2 arguments
  - E.g., "Is 1 the square of 1?" "Is 4 the square of 2?" etc.
  - Cf. "What is the square of what?"
  - $\{(1, 1), (2, 4), (3, 9), ...\} = \{(x, y) \mid y = x^2\}$
- 1 argument
  - E.g., "Is 1 a square?" "Is 4 a square?" etc.
  - Cf. "What are squares?"
  - $\{1, 4, 9, ...\} = \{x \mid x \text{ is a square of some number}\}$

Computational Problem

- Checking the membership of a set (could be a relation)
- Then, we may just identify a computational problem with the *set*. ~ *language* Recall this?
- Examples (cf. Group Exercise 1)
  - {*Mercury*, *Venus*, *Earth*, *Mars*, *Jupiter*, ... }
  - {(*Nobo*, *Penang*), (*Furby*, *home*), ...}
  - {2005, 2006, 2007, 2008, 2009, 2010, ...}

Potential problems with this description?

Interim Summary: Problem Transformation

- Practical problem ~ action
  - Significance: Real-life situation
- Research problem ~ question
  - Significance: possibility of solving the associated practical problem (if acted)
- Computational problem ~ set
  - Significance: Abstract/computational treatment ⇒ Only yes-no response needed

Practice: "Dating"

- Practical problem: Date an ideal partner.
- Research problem:
  - Cost/significance
- Computational problem (set):

Group Exercise 2

- Transform the following research problems into the corresponding computational problems, and represent them as sets

1. Is it going to rain this evening?
2. Why do people (still) smoke cigarette?
3. How can one cook pasta?

If you need clarification, ask questions

Group Exercise 3

- Identify each of your own problem (Ex 00) with respect to the following stages:
  - Practical
  - Research
  - Computational
  - Set
- Transform your problems so that you eventually convert them into sets (may need to make adjustments)

Work on your problems **jointly within your group**

Unit Summary

- Types of problems: practical, research, computational
  - Why categorize?
- Computational problem ~ set
  - Why set?
- Problem transformation
  - How to do it?

Cf. Content Goal 1

Exercise A1: Using a Theory

- Part 1: Using a Theory
  - Task 1: Identify theories (in/out CS)
  - Task 2: Define “theory” (your own)
  - Task 3: Personal feelings about theory
- Part 2: Review “Problems”
  - Group Exercise 3 (previous slide)
- Unit A1 Summary question
  - Were you sufficiently motivated to discuss “problems” as the starting point of this course? Explain.