Unit D6: Overview

- Explore models of super-Turing computability
  - Reading: van Leeuwen & Wiedermann
  - Other models
- Summaries: Unit, Course
- Exercise D6 (Module D Comprehensive) "Mini Research"

Group Exercise 1

- Compare your exercises (reading van Leeuwen & Wiedermann)
- Prepare to present interesting points (possibly including similarities and differences)

Reading: van Leeuwen & Wiedermann

1. What is the main research question of the paper?
2. What is the significance of the research question?
3. Does the paper respond to the research question well?
4. Is the paper organized well?
5. Summarize the author's arguments *in the order of importance* (as you understand).
6. Were you convinced by Proposition 5 and its proof (p. 10)?
7. What is the difference between the (traditional) Church-Turing thesis and the extended Church-Turing thesis (proposed in the paper)? What kind of impacts could the extension have?
8. Try to criticize the author's position/arguments. If you cannot do this, explain why.
9. How would the author's arguments affect your analyses of your own problems and/or your choice of the mini research problems that have been done so far, *focusing on computability and (time) complexity*?

Sample Problems

- Driving from one place to another
- "Real" web programming (arachnidan)
- Bird flocking
- Psycho-social behaviors
  - E.g., child-parent attachment, language
- "Gaia" hypothesis (the entire earth as an organism)
- The universe

cf. Chomsky

Can site machines/TM with advice do these?

Super-Turing: Requirements

- Interactivity: Can accept input and deliver output in the middle of computation
- Robustness: Can deal with fuzzy, unknown, unpredictable environments (inputs)
- Adaptability: Can change according to the environment (i.e., learning)

Super-Turing: Assumptions

- Extension: The TM-based "theory" is to be a special case of the theory of interaction.
- Multiple levels of representation (cf. Church-Turing Thesis)
  - Machine models (cf. TMs, URMs, RAMs)
  - Mathematical/functional models (cf. recursive functions, λ-calculus)
- Ockham's Razor: Choose the simplest if multiple options are available

Why this many?

Super^{TM} Models

- Site machines (concrete machine model) [van Leeuwen & Wiedermann]
- TMs with advice function (abstract machine model) [van Leeuwen & Wiedermann]
- Interaction machines (persistent TMs) (abstract machine model) [Wegner & Goldin]
- yet another machine model (intentionally kept untold at this point)
- π-calculus (functional model) [Milner and others]
- Neural models [e.g., Siegelmann]
- Quantum/relativistic computing

cf. on-line references

Interaction Machine (IM)

- Note: TM computation = Single session
- IM computation = Multiple sessions
- Interactivity: Realized as *repeated* TM sessions where information can be *stored* on a working tape
- Multiple input streams on multiple tapes
- An extension of a multiple-tape TM

Variations?

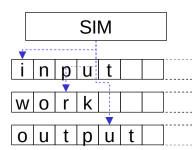
Variations

- General: MIM (Multiple-stream IM)
  - Model of distributed computing
- Special: SIM (Single-stream or Sequential IM)
  - Model of 2-agent interaction

MIM: not yet well-developed

Single-stream IM (SIM)

- Single input stream on a single tape
- Work tape will be preserved (persistent).
- Behavior: set of I/O streams (cf. TM behavior: set of I/O pairs)



Example: Answering machine
(input, work) => (output, work)
- (record Y, X) => (ok, XY)
- (playback, X) => (X, X)
- (erase, X) => (done, ε)

Properties of SIMs

- SIM states depend on the content of the working tape [no bound].
- Infinitely many states (cf. finite TM states)
- Uncountably many SIM's (cf. countably many TMs)
- Can handle uncountable sets (cf. TM can handle countable sets, i.e., uncountable ~ unsolvable)

Properties of MIMs

- No single observer (on a single stream) has complete information.
  - Cf. A SIM has complete information (about the interaction between the two parties).
  - Analogy: "Blind men and an elephant"
- Multiple streams cannot be serialized.

Example: Delegation
- Stream 1: (main-in, main-out)
- Stream 2: (sub-out, sub-in)
- Coordination: (main-in, sub-out, sub-in, main-out)

Comparison

- TM: 1-agent systems ⇔ countable sets
  - Traditional Church-Turing Thesis for algorithmic computability (TM, λ-calculus)
- SIM: 2-agent systems ⇔ uncountable sets
  - Extended Church-Turing Thesis for sequential interaction (SIM, calculus?)
- MIM: *n*-agent systems ⇔ ?
  - Ultimate Church-Turing Thesis for general interaction (MIM, calculus?)

TM as amnesic SIM

Analysis of IM

- Interactivity
- Robustness
- Adaptability

Functional Model

- TM ⇔ λ-calculus
- SIM ⇔ ?
- MIM ⇔ ?

λ-calculus

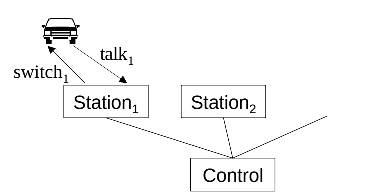
Formal representation of functions

- Abstraction: $\lambda x.exp$ [define a function]
- Application: $exp_1\ exp_2$ [give an argument]
- β-reduction: $(\lambda x.exp)\ y$ [use a function]
  - The result is *exp* where the instances of *x* is replaced with *y*.
  - E.g., $(\lambda x.f(f(x)))\ 0 = f(f(0))$

λ-calculus Examples

- Identity function: $\lambda x.x$
- Church numeral
  - Representation of a natural number *n*: $\lambda f.\lambda x.f\ n(x)$
  - $0 = \lambda f.\lambda x.x$ [Note: $\lambda f.\lambda x.f\ 0(x)$]
  - Succ = $\lambda n.\lambda f.\lambda x.(f\ ((n\ f)\ x))$
  - 1 = Succ 0 = $\lambda f.\lambda x.(f\ (((\lambda f.\lambda x.x)\ f)\ x)) = \lambda f.\lambda x.(f\ (\lambda x.x\ x))$ $= \lambda f.\lambda x.(f\ x)$ [Note: $\lambda f.\lambda x.f\ 1(x)$]
- Conditional
  - T = $\lambda x\lambda y.x$ [i.e., picking the first argument]
  - F = $\lambda x\lambda y.y$ [i.e., picking the second argument]
  - B M N (if B then M else N) where B is T or F

Example Scenario



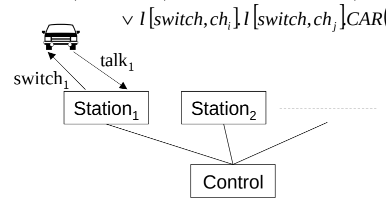
π-Calculus

Changes to λ-calculus

- Output mechanism
  - Cf. β-reduction: $(\lambda x.exp)\ y$ as an input mechanism
- Parallel composition of processes
  - Commutative, associative
- Meaning as state change
- Referencing (by name)
  - Cf. pointers in modern programming languages

π-Calculus Example

$$CAR(talk, switch) \overset{def}{\equiv} out\,[talk, word].CAR(talk, switch) \vee l\,[switch, ch_i].l\,[switch, ch_j].CAR(ch_i, ch_j)$$



$$NETWORK \overset{def}{\equiv} \underbrace{(CAR(talk_1, switch_1) \mid STATION_1 \mid \ldots \mid CONTROL)}_{talk_1, switch_1}$$

simplified notation

Analysis of π-Calculus

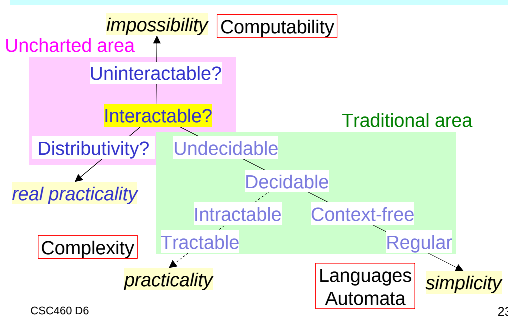
- Interactivity
- Robustness
- Adaptability

Super-Turing Thesis

- Would all models of interaction converge?
- Would it be the next level of computation/interaction/emergence?

Both Turing and von Neumann have foreseen super-Turing computation, even though their names are more strongly associated with algorithmic computation. Why did most computer scientists pick up only the algorithmic aspect of their ideas?

Extended Theory of Computation



Unit Summary [critical view]

- TM-based theory
  - Stimulus-response without thinking
  - Vertebrate with a disabled brain
    Cf. "chicken without a head"
- Theory must evolve into a more sophisticated beast.
  - Modeling full-fledged interaction
- Some building blocks are emerging.

Preview

Module D Evaluation Workshop

- Mini research presentations
  - Know the time limit; Think big; Be creative
- Peer discussions
- Reflection and self-evaluation

Practicum evaluation
- Wed., Apr. 27
- Detailed instructions will be distributed later (but already available on-line)

Course Summary

- Understanding the “Theory” ⇒ Computer scientist (good at solving algorithmic problems)
- Understanding the limitations of the “Theory” ⇒ Philosopher (good at solving “real” problems)
- Change in your brain?

Sin minas, no hay tango.
Sin teoría, no hay ciencia.

Surveys