Name: ______________________

Exercise D5, 4/19/05

Reading: van Leeuwen & Wiedermann

After reading and discussing MacLennan, we may or may not agree with all of his points. However, one point we cannot overlook is that there is a growing number of researchers who take a position similar/related to MacLennan (see the course reference list for more information; Copeland [2003] is a detailed review). In order to gain additional insight into the area beyond TMs, we will read a paper by Jan van Leeuwen and Jirí Wiedermann, who propose a "super-Turing" thesis to capture the notion of computation at a higher level, potentially more applicable to modern computing.

Reference: van Leeuwen, Jan and Wiedermann, Jirí. 2001. The Turing machine paradigm in contemporary computing. In *Mathematics Unlimited - 2001 and Beyond*, eds. B. Enquist and W. Schmid, 1139-1155. Springer. [the distributed copy, available on-line, is a draft version of (i.e., not necessarily identical to) the listed paper; van Leeuwen's web site: http://www.cs.uu.nl/people/jan/]

Task 1: Read the paper carefully. Try to relate the author's argument with what you learned in this course. Note: You may skip the details of certain technical parts, e.g., the "advice" function.

Task 2: Concisely write up your response to the following questions

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*?

Be prepared to discuss your response in class.

Survey: Time spent between classes: ____________

Notes for the reading

- Oracle computation (p. 4): If a TM is given an oracle that decides some undecidable problem (set), then the TM could recognize a non-TM-recognizable language. Naturally, the increased power in this case comes from the power of such an oracle, which is beyond a TM.
- Advice (p. 6): A limited version of oracle (because an oracle could be excessively powerful). Discussed more in detail in Section 3.2.1.

- Mapping Φ: $(\Sigma^k)^\infty \rightarrow (\Sigma^l)^\infty$ (p. 7): A function whose input is an infinite sequence of *k* symbols on *k* ports and whose output is an infinite sequence of *l* symbols on *l* ports. As an example, suppose a machine with a keyboard and mouse as input device and display and printer as output device. Imagine that the keyboard deals with symbols from the set *K*; analogous for other devices: *M* for mouse, *D* for display, and *P* for printer. Then, the mapping/function would be as follows: Φ: $(K \times M)_{\text{[for time 1]}} \times (K \times M)_{\text{[for time 2]}} \times \ldots \rightarrow (D \times P)_{\text{[for time 1]}} \times (D \times P)_{\text{[for time 2]}} \times \ldots$
- Advice function (p. 8): The authors consider that oracle computation without limit is too powerful (p. 4). What they propose is to use a limited version of oracle called "advice." Their "advice" takes the form of function: given a natural number, the function returns a string. The power of the function is basically "polynomial," but include certain "randomization" possibilities, useful for reflecting unknown future. Here, we consider this function as a "reasonable" way to integrate mostly tractable but not completely predictable. Some additional notes are included below (indented).
  - Bounded by $S(n)$ (p. 8): $S(n)$ refers to "space bound" with respect to the input size (cf. $T(n)$ for "time bound). Note that this bound applies to the F part of the class specification "C/F" (see below).
  - Class C/F (p. 8): The main motivation for this definition is to introduce the class *P*/*poly* (see below). Skip the details.
  - Class *EXPTIME* (p. 8): This corresponds to the class Exp used in this course.
  - Class *poly* (p. 8): The class of polynomially (size-)bounded advice function. Recall that the space bound $S(n)$ applies to F, and *poly* is one case of F.
  - Class *P*/*poly* (p. 8): Roughly, this class is a "non-uniform" (evolutionary, upgradable, etc.) analogue of *P*. It contains *P* (i.e., $P \subset P/poly$) as well as some undecidable sets (due to its upgradability, which is not fixed at the beginning). The comparison with *NP* is not yet known. The fact that *P*/*poly* contains *RP* and *BPP* (see below) suggests that *P*/*poly* captures some randomized behavior.
  - Class *NP*/*poly* (p. 8): Ignore this class.
  - Class *RP* [Randomized Polynomial Time] (p. 8): A language is in *RP* if there is a boolean verifier computable in deterministic polynomial time (or finding a boolean satisfiability answer with a NTM) such that the input is accepted at least 1/2 of the time.
  - Class *BPP* [Bounded-Error Probabilistic Polynomial Time] (p. 8): Analogous (or more confident version) to RP, except that the input must be accepted at least 3/4 of the time (the value could be others, e.g., 2/3, as long as it is strictly greater than 1/2, as they can approach 1 by repeated multiplications).
- ω-automata (p. 9): Variants of FSAs that accept infinite strings (not just potentially infinite as in the case of FSAs).
- Non-recursive (p. 10): Beyond recursive, i.e., r.e. (TM-recognizable) or non-r.e. (non-TM-recognizable).
- $\langle M \rangle$ (p. 10): A standard way of indicating that this is a representation of a TM. We could consider this as the TM's formal definition, as discussed in class.
- Proof of Proposition 5 (p. 10): The basic idea is that the advice function identifies the maximum number of steps for a TM whose representation is of some length. Although the advice function adds power to the TM, it is still within the bound. So, it is not considered unreasonably powerful. However, we know that such a function is not decidable by a standard TM.
- Theorem 6 (p. 11): Ignore it.
- S = (γ) (p. 11): Here the site machine is defined (as a 1-tuple) in terms of the function γ from time to a hardware/software description (p. 7).
- Instantaneous description (p. 11): A snapshot of a machine. The instantaneous description of a TM (or similar machine) includes the current state, head position, and the tape symbols. Note that if *n* tape positions are used, there would be at most $|\Gamma|^n$ symbol possibilities, where Γ is the set of tape symbols; this suggests that it would have taken at most exponential time to have used the space.
- Infinite circuit families (p. 16): Ignore this.
- BSS model (p. 16): After Blum, Shub, and Smale. The problem of deciding general existential formulas in the first-order theory over the reals, which is a real analogue of NP-complete.
- Neuroidal networks (p. 16): A specific version of a neural networks, proposed by a computational learning expert Valiant.

// End