

On Programming Languages

Due — Tuesday, May 1, 2001 (midnight)

I. A Continuous Programming Language

- A. Current programming languages are digital in almost every sense. Nevertheless, many of them—at least at some level of idealisation—support non-digital phenomena in various senses. For example, we are entirely used to programming language support for real number values.

Suppose you were to extrapolate from that simple example and take on the project of designing a **continuous programming language**. Imagine, for example, allowing real-valued indices for arrays—so that you could allocate an array $A(i)$ for i from 0.0 to 1.0. It would no longer be possible to *step through* such an array, of course—so one might want to add a continuous SCAN operator, so as to support something like the following:

```
begin scan i from 0.0 to 1.0
  if  $a(i) > \max$  then  $\max \leftarrow a(i)$ 
end
```

- B. Setting aside all specific considerations of implementation for a moment—but without spilling over into behaviours that in some pretheoretic sense would require an oracle (i.e., things that are actually *metaphysically* non-effective)—suggest other continuous constructs that might be useful: data structures, control structures, whatever. **Be obstreperous!** Let your imagination run wild. See how far you can get in eliminating our present habits of discreteness from programming languages altogether.
- C. Briefly discuss the issues that would be involved in implementing your proposal. Is it metaphysically challenged? physically precluded? possible but ad-hoc? possible but unreliable? inherently infinite? approximable on a discrete machine? perfectly implementable?
- D. Note: this is an extremely open-ended assignment. In the past, just to give you an idea of the space of possibilities, students have written about continuous extensions or extrapolations of such familiar programming language constructs as: trees (with their traditionally discrete branching factors); threads; and recursion (what would it be to recurse to a depth of 1.46?). You are welcome to explore any of these possibilities—or any other that strikes your fancy.