Lecture #9: Nondeterministic Time — Speedup, Emulation, and a Nondeterministic Time Hierarchy Theorem

Lecture Presentation

This lecture describes a surprisingly simple simulation of a k-tape nondeterministic Turing machine

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$$

by a three-tape nondeterministic Turing machine, with input alphabet $\widehat{\Sigma} = \Sigma \cup \{\#\}$ (for a symbol # that does not belong to Σ) — whose correctness might not be obvious. This algorithm processes a string $\mu \in \widehat{\Sigma}^\star$ which should encode both an input string ω for M, and a bound T on the number of steps that M should take. It is summarized in Figure 2 (which can be found several pages after this one).

As noted above, it might not be obvious that this algorithm is correct — so it might be helpful to consider a *different* process, which uses $k+2$ tapes, that is shown in Figure 1, before that.

The correctness of the simulation given in the lecture notes can be established as follows:

1. Establish the correctness of the simulation that is summarized in Figure 1.
2. Establish that if the simulation in Figure 1 is correct, then the simulation in Figure 2 must be correct too.
3. Conclude that the simulation in Figure 2 is correct.

Note: The simulation, being considered here, is adapted (in the lecture notes) to describe a "nondeterministic universal Turing machine" which decides a language $A_{\text{NTM+I+Time}}$. This Turing machine (and results that can be proved about it) is then used to establish a "Nondeterministic Time Hierarchy Theorem" — so its correctness is (arguably) important.

```
On input μ ∈ Σ̂* {
  if (μ is ω#^T for a string ω ∈ Σ* and a positive integer T) {
    Nondeterministically guess a sequence χ_1, χ_2, ..., χ_t of possible moves,
    for an integer t such that 1 ≤ t ≤ T
    for (1 ≤ i ≤ t) {
      if (initial state in χ_i makes sense) {
      for (1 ≤ j ≤ k) {
        if (management of Tape #j by χ_i does not make sense) {
          reject
        }
      }
    }
    if (χ_t take M to state q_accept) {
      accept
    } else {
      reject
    }
  } else {
    reject
  }
}
```

Figure 1: Another simulation of a k-Tape Nondeterministic Turing Machine

Correctness of the Simulation in Figure 1

On input $\mu \in \widehat{\Sigma}^\star$ {

1. `if` (μ is $\omega\#^T$ for a string $\omega \in \Sigma^\star$ and a positive integer T) {
2. Nondeterministically guess a sequence $\chi_1, \chi_2, \ldots, \chi_t$ of possible moves, for an integer t such that $1 \le t \le T$
3. `if` (states in $\chi_1, \chi_2, \ldots, \chi_t$ make sense) {
4. `for` $(1 \le j \le k)$ {
5. `if` (management of Tape #j by $\chi_1, \chi_2, \ldots, \chi_t$ *does not* make sense) {
6. `reject`

   }
7. `accept`

   } `else` {
8. `reject`

   }

   } `else` {
9. `reject`

   }

}

Figure 2: Nondeterministic Simulation of a k-Tape Nondeterministic Turing Machine

Correctness of the Simulation in Figure 2