ECM3404: Logic and Computation

Tutorial, 20th March 2009

Turing Machines

1. Construct a *copier* for strings over $\{0, 1\}$, i.e., a Turing machine meeting the specification:

   $$\text{Initial configuration:} \quad \cdots \Delta \underset{1}{\boxed{a_1}} a_2 \cdots a_n \Delta \cdots$$
   $$\text{Halting configuration:} \quad \cdots \Delta a_1 \cdots a_n \Delta \underset{0}{\boxed{a_1}} a_2 \cdots a_n \Delta \cdots$$

   where $a_i \in \{0, 1\}$ for $i = 1, 2, \ldots, n$. Use as many extra tape symbols as you want. You should be able to do this using at most six non-halting states.

2. Now construct a Turing machine with the same functionality, but with no extra tape symbols (so $\Gamma = \{0, 1, \Delta\}$). (I can do this using nine non-halting states—maybe you can do better.)

3. Design a *unary multiplier*, i.e., a Turing machine meeting the specification:

   $$\text{Initial configuration:} \quad \cdots \Delta \overbrace{\underset{1}{\boxed{1}} \cdots 1}^{m} \Delta \overbrace{1 \cdots 1}^{n} \Delta \cdots$$
   $$\text{Halting configuration:} \quad \cdots \Delta \overbrace{\underset{0}{\boxed{1}} \cdots 1}^{mn} \Delta \cdots$$

   See if you can do this without using any extra tape symbols.