Turing Machines and Simulations

CS154

Chris Pollett

Apr 23, 2007.

Outline

- Equivalence classes of automata
- Stay put machines
- Semi-infinite tape
- Offline Turing Machines

Equivalence classes of automata

- Given two machine models F and F´ we say they are **equivalent** if for each M in F there is an M´ in F´ such that L(M) =L(M´).
- We are interested in what models are equivalent to TMs.

Stay put machines

- Suppose rather than having a transition function $\partial : Q \times \Gamma \dashrightarrow Q \times \Gamma \times \{L,R\}$ we instead have: $\partial : Q \times \Gamma \dashrightarrow Q \times \Gamma \times \{L,R,S\}$ where S denotes stay put.
- It is obvious that machines with this extra stay put ability can simulate machines that don't have it.
- On the other hand, given a transition:

$$\partial(q,a) \dashrightarrow (r, b, S)$$

- We could simulate it by having transitions:

$$\partial(q,a) \dashrightarrow (r', b, R)$$

$$\partial(r',c) \dashrightarrow (r, c, L)$$

where r´ is a new state and where we have the second kind of transition for each symbol c.

- Hence, stay-put machines are equivalent to usual TMs

Semi-infinite tape

- Suppose rather than having a two way infinite tape we instead have a tape which has is infinite only to the right and where the machine starts off on the left hand square of the input.
- If the machine ever tries to move off to the left hand side of the tape it just stays on that left hand square.
- This can be simulated by a usual TM by having our machine as follows: (1) for each tape symbol a we have a new symbol $\underline{a}$ which means we are on the left hand side of the tape reading an a. (2) The first move of our machine replaces whatever symbol a it was reading with $\underline{a}$. Thereafter, our machine acts like a usual TM. For each transition which moves right ∂ (q,a) -->(r, b,R) we also add a transition ∂ (q,$\underline{a}$) -->(r, $\underline{b}$, R). For each transition that moves left ∂ (q,a) -->(r, b, L) we add, ∂ (q,$\underline{a}$) -->(r, $\underline{b}$,S).

Semi-infinite tape machines can simulate usual TMs.

- The idea is to again increase the tape alphabet Γ. Let Γ' be $\Gamma \cup \{\underline{a} \mid a \text{ in } \Gamma\}$, and let the new alphabet be $\Gamma' \cup (\Gamma' x \Gamma')$. This is still finite.
- The simulator acts by first replacing the input $w_1,\ldots,w_n$ with $(\underline{w_1}, \_)$, $(w_2, \_) \ldots (w_n, \_)$.
- Now we have for each original transition a transition which acts on the left hand side of a symbol.
- We also have transitions which when we move left in a position with an underscore in left coordinate we "move" onto the right hand coordinate. I.e, we can start affecting it. Then for transitions in the original machine to the left again we would have transitions which move right on the right hand coordinate and vice versa. The right hand cordinate thus, behaves as the squares in the original machine to the right of the first square of the input.

Off-line Turing Machines

- Offline Turing Machines are TMs with two tapes.
  - They have an input tape which is read only on which the input is originally written.
  - They have a work tape which is initially blank but which is read write.
- Offline machines can simulate usual machines by first copying the input tape to the work tape and then doing all further work on the work tape like a usual TM.
- The basic idea is to introduce a new symbol # to denote the stop and end of tape configuration we also have underscore for each tape symbol. If the input to the offline machine is $w_1 \ldots w_n$, our simulator first converts it to the single string $\# \underline{w_1} w_2 \ldots w_n \# \underline{-} \#$. Between the first two #'s represents the contents of the input tape, between the second two represents the contents of the work tape that we might have affected (initially, only the square we start at). The underscores indicate which square is being read.
- To simulate one step of the offline we scan our tape left to right to determine which symbols are under the tape head. Then we scan right to left to update the tapes according to the offline tape's transition.