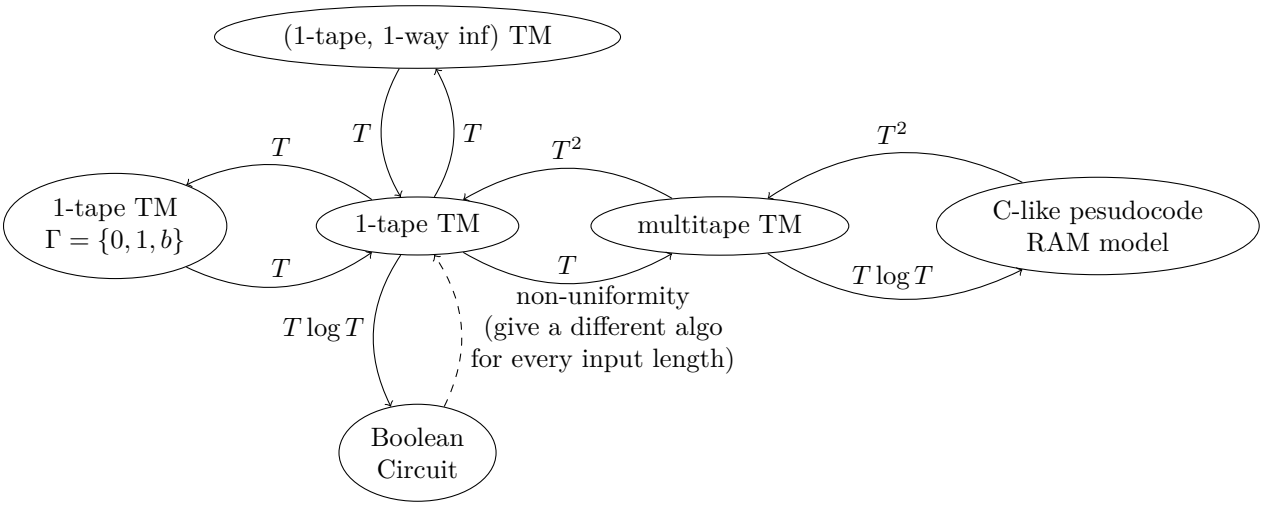
Undergraduate Complexity Theory
Lecture 3: Simulation and Turing Machine Variants

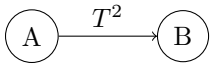
Marcythm

July 9, 2022

1 Lecture Notes

Somethings we are going to prove or talk about today:



Definition 1.1. 

means can "compile" (simulate) code M_A in model A to code M_B in model B, s.t. if M_A runs in time T, then M_B runs in time $O(T^2)$.

Definition 1.2. For a **decider** M, the *running time (time complexity)* of M is a function $T : \mathbb{N} \to \mathbb{N}$:

$$T(n) = \max_{|x|=n} \{\#\text{ steps } M(x) \text{ takes}\}$$

Remark 1.3. Time Complexity is a **function** of n, because we care about how it **scales** with input size.

TM tricks:

1. Allow TMs to 'S'tay and put in a step besides L/R.
2. LL/RR in the same way.
3. "Marking" a cell. Impl: just double the tape alphabet Γ.
   Simulate 2-way infinite TM with 1-way infinite TM: using marking trick to indicate left boundary.
4. "Stretching" an input, i.e. $abcc \to a\_b\_c\_c$.
   Simulate $\Gamma = \{0, 1, b, \dot{0}, \dot{1}, \dot{b}, \#\}$ with $\Gamma = \{0, 1, b\}$: use stretch to have room for new encoding.
   Simulate multitape TM with 1-tape TM.

2 Reading

2.1 Sipser 3.2 (Variants of Turing Machines)

1. multitape TM, and how to simulate it

2.2 Sipser 7.1 (Measuring Complexity)

1. time complexity of TM
2. asymptotic notations
3. difference between:
   (a) computability theory: all reasonable models of computation are equivalent, i.e. they all decide the same class of languages.
   (b) complexity theory: the choice of model affects the time complexity of languages.