CPSC 500
Lecture 1: Sept. 4, 2013

Scribe: Rebecca McKnight

1 Overview

In this lecture, computational models were discussed in order to give a sense of how to quantify various algorithms in terms of space and time. For example, what is meant by $O(n^3)$ time and $O(n \log n)$ space? A computational model is needed to provide context for such a question.

Computations on matrices, for example, could include:

- counting row operations (measuring *time*)
- finding the size of a matrix (measuring memory used, or *space*)
- counting FLOPS

Computations on graphs, for example, could include:

- counting the number of nodes (measuring the *input size*)

Each of these concepts (time, space, and input size) are the key features for quantifying algorithms, and are given context by a computational model.

2 Computational Models

2.1 Turing Machines

A Turing Machine, named for famous computer scientist Alan Turing, is a very basic yet powerful computational model. This machine works by moving a tape head right or left along an input tape, one cell at a time, and performing operations based on a transition function. The key components of a Turing Machine are as follows: [1]

- The input tape, which is broken up into cells that each contain exactly one character or symbol from a certain alphabet, which includes a "blank" symbol. In general, the tape is considered to have a leftmost cell, whereas the tape continues infinitely to the right.
- The tape head, which moves left or right one cell at a time, and can read from and write to the input tape.

- The state register, which keeps track of the current state of the Turing Machine, and is initialized to a certain start state.

- The transition function, which gives instructions to the machine based on its current state. Instructions follow the sequence:

  1. Write a symbol (which could be the "blank" symbol)
  2. Move the tape head right or left
  3. Either stay in the current state, or change to a new state

Every computation that can be performed on a modern computer can be broken down and computed on a Turing Machine, hence why it is such a powerful model. However, it is not necessarily the most realistic model in terms of complexity. Since the tape head can only move to an adjacent cell, and can only move one cell at a time, evaluating the *time* of an algorithm will not be an accurate representation of reality.

2.2 Random Access Machines

A Random Access Machine, or RAM, is a more realistic computational model which provides a more realistic context for the evaluation of computations. A RAM is roughly equivalent to a Turing Machine, however it has a few key differences which not only make it more realistic, but also make it easier to program and use.

The defining attributes of a RAM include:

- The ability to move to any location using only one step
- A finite number of registers, which can store values for later retrieval
- New operations "add," "store," and "recall"

Since the RAM allows for easy access to the input and stored data, it gives us a more flexible and realistic model of computation, since these operations more closely match those of modern computers.

2.3 An Example - Palindromes

A palindrome is a word or phrase (although spaces and punctuation are not taken into account) that can be written the same backwards as it is forwards. For example, "MADAM I'M ADAM" is a palindrome, since the letters "MADAMIMADAM," when written in reverse, have the same ordering. Determining whether a string is a palindrome or not is a common example of a computational problem.

To solve the palindrome decision problem using a Turing Machine, it would take $O(n^2)$ time since for each character, the head needs to travel to the other end of the string to ensure the correct character appears. This running time is not ideal, nor is it realistic.

In contrast, the palindrome decision problem takes only linear time to solve using a Random Access Machine, since the head can jump around from one character to its corresponding character in the second half of the string using only one step. This is a much more accurate representation of reality.

3 Conclusion

Although this lecture was a quick introduction to the concept of computational models, we already get a sense of their importance in evaluating algorithms. Without such models, the ideas of time and space are undefined, which makes them useless for our purposes. However, within the context of a model, we can properly evaluate and compare different algorithms. Even though they are often much simpler than modern computers, models such as the Random Access Machine give us a feeling for what space and time mean, even for complex computations.

References

[1] Wikipedia, "Turing machine," http://en.wikipedia.org/wiki/Turing_machine.