Giorgi Japaridze

Theory of Computability

Parallel computation

Section 10.5

Introduction

A ***parallel computer*** is one that can perform multiple operations simultaneously. Such computers may solve certain problems much faster than ***sequential computers***, which can only do a single operation at a time.

In practice, the distinction between the two is slightly blurred because most real computers (including "sequential" ones) are designed to use some parallelism as they execute individual instructions (remember pipelining after all). We focus here on *massive* parallelism whereby a huge number (think of millions or more) of processing elements are actively participating in a single computation.

One of the most popular models in theoretical work on parallel algorithms is called the ***Parallel Random Access Machine*** or ***PRAM***. In the PRAM model, idealized processors with a single instruction set patterned on actual computers interact via a shared memory.

Our textbook, however, uses an alternative, simpler model of parallel computers. Namely, Boolean circuits, already seen in Section 9.3.

Uniform Boolean circuits as parallel computers

In the Boolean circuit model of a parallel computer, we take each gate to be an individual processor, so we define the ***processor complexity*** of a Boolean circuit to be its *size*. We consider each processor to compute its function in a single time step, so we define the ***parallel time complexity*** of a Boolean circuit to be its *depth*.

Any particular circuit has a fixed input size (= number of input variables), so we use circuit families as defined in Definition 9.27 for recognizing languages.

We however need to impose a technical requirement on circuit families so that they correspond to parallel computation models such as PRAMs where a single machine is capable of handling all input lengths. That requirement states that we can easily obtain all members in a circuit family. This *uniformity* requirement is reasonable because knowing that a small circuit exists for recognizing certain elements of a language isn't very useful if the circuit itself is hard to find. That leads us to the following definition.

Definition 10.34 A family of circuits $(C_1, C_2, \ldots)$ is ***uniform*** if some log space transducer **T** outputs $\langle C_n \rangle$ when **T**'s input is 1^n.

We say that a language has ***simultaneous size-depth*** circuit complexity at most $(f(n), g(n))$ if a uniform circuit family exists for that language with size complexity $f(n)$ and depth complexity $g(n)$.