CS411 Theory of Computation
Lecture 1

Sergey Kitaev

University of Strathclyde

20 September 2017

CS411 Logistics

- 20 Lectures on Wednesdays at 10-12 in MC 303.
- There will be intermittent tutorials for this course on Thursdays at 11-12 in GH 898. These are to be announced on lectures.
- Marking scheme: Exam (70%), two homework assignments (15% each).
- Two homeworks:
  HW1 due Wednesday, October 18th, 12pm (5th week);
  HW2 due Wednesday, November 22rd, 12pm (10th week).
- Note on end-of-year exam format.
- Course material is available on MyPlace.

Course content

Part 1:

- Turing machines and algorithms
- Turing recognizable and decidable languages
- Variants of Turing machines
- Equivalence of non-deterministic and deterministic TM's
- Recognizability of the Halting problem; undecidability of the Halting problem

Course content continued

Part 2:

- Complexity, Big O and little o notation, program analysis and complexity classes. Complexities of TM's.
- Comparing the complexity of deterministic and non-deterministic TM's.
- Polynomial time, the class P.

Part 3:

- The class NP, examples of problems in NP.
- SAT and the Cook-Levin theorem
- SAT is NP-complete.

Background reading/support material

Introduction to the Theory of Computation (2nd edition)

Michael Sipser

Motivation

- What are the fundamental capabilities and limitations of computers?
- What makes some problems computationally hard and others easy?
- In addressing these questions we require some notion of a machine to model computation.
- Such a machine should be be able to do all those things a Computer (in the modern sense of the word) can do.
- From such a model we can begin to build a theory of computation.

Some important points

- From the title of this course, the content of the material will be quite theoretical. So do expect many Definitions, Theorems, etc.
- While there is no escape from this please do not be put-off by it.
- In order to describe our theoretical machine, we will need to recall some basic discrete objects such as sets, graphs and sequences. With these in place we can begin to describe (accurately) what our machine will be and do...
- *** While these slides contain SOME of the course material, other material may be done on the board.

  It is therefore ESSENTIAL that you show up to class. The material done on the board will NOT be uploaded.

Sets

- A set is a collection of objects. The objects in a set are called its elements/members.
- Sets are written with braces, e.g. $\{5, 10, 15\}$ is a set containing three elements, e.g. $\{\texttt{apple}, \texttt{banana}, \texttt{mango}, 1\}$ is a set containing 4 elements.
- The symbols $\in$ and $\notin$ denote set membership and non-membership, e.g. $4 \in \{3, 4, 6\}$ but $\texttt{D} \notin \{3, \texttt{L}\}$.
- The empty set is denoted $\emptyset$.
- $A \subseteq B$ means A is a subset of B, i.e. every element in A is also in B.
- Two common sets:

$$\mathbb{N} = \{1, 2, 3, 4, \ldots\} \qquad \text{the set of natural numbers}$$
$$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\} \qquad \text{the set of integers.}$$

Sequences and Tuples

- A sequence is a collection of objects written in a certain order.
- Sequences are usually written as a list within parentheses.
- e.g. $(5, 2, E, W)$, (u, d, l, r).
- Finite sequences are often called tuples.
- A sequence with k elements is a k-tuple. A 2-tuple is called a pair.
- Sets and sequences may appear as elements of other sets and sequences.
- The power set of a set A is the set of all subsets of A. For example, if $A = \{x, y\}$ then

$$\texttt{power set}(A) = \{\emptyset, \{x\}, \{y\}, \{x, y\}\}.$$

Cartesian products

- Given sets A and B, the Cartesian product (or cross product of A and B) is the set of all pairs (a, b) where $a \in A$ and $b \in B$. This is written as
$$A \times B = \{(a, b) | a \in A,\ b \in B\}.$$
- e.g. If $A = \{3, t\}$ and $B = \{f, 5, x\}$ then
$$A \times B = \{(3, f), (3, 5), (3, x), (t, f), (t, 5), (t, x)\}.$$
- The Cartesian product of the k sets $A_1, A_2, \ldots, A_n$ is
$$A_1 \times A_2 \times \cdots \times A_k = \{(a_1, a_2, \ldots, a_k) | a_1 \in A_1, ..., a_k \in A_k\}.$$
- The Cartesian product of a set with itself k times is
$$\overbrace{A \times A \times \cdots \times A}^{k} = A^k$$

Functions

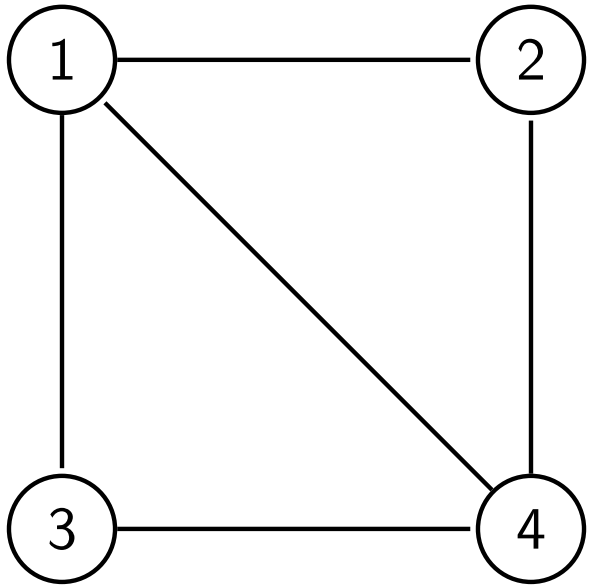
- Functions can be defined is several (equivalent) ways. A function from a set A to a set B is usually written as $f : A \to B$.
- One way to think of a function is as something that assigns an element of B to every element of A. If f acts on a and 'outputs' b then we write $f(a) = b$.
- Another way to think of a function is as a subset of the Cartesian product $A \times B$. If $(a, b) \in f$ then this means that $f(a) = b$.
- Given $f : A \to B$, the set A is called the domain of f, and B is called the range of f.

More functions

- When the domain of a function is a cross product of k sets, then we say the function is a k-ary function. An example of a 2-ary function is addition: $+ : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$.
- A 2-ary function is more commonly called a binary function.
- A predicate is a function whose range is $\{\text{True}, \text{False}\}$.
- Other properties of functions will be recalled as they are needed.

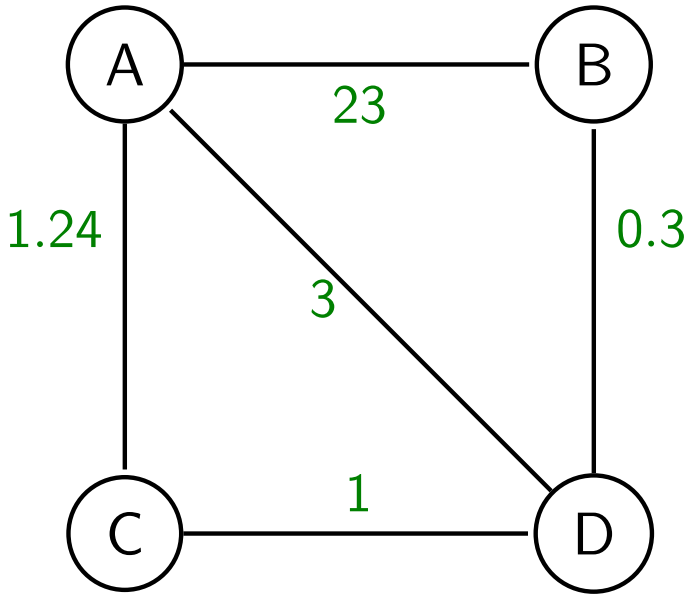
Graphs

An (undirected) graph G is a set of points (vertices/nodes) and a set of edges which summarize which points are connected. G is written as a pair (V, E) where V is the set of vertices and E is the set of edges.



For this graph, $V = \{1, 2, 3, 4\}$ and $E = \{(1, 2), (1, 3), (1, 4), (2, 4), (3, 4)\}$.

Labelled graphs



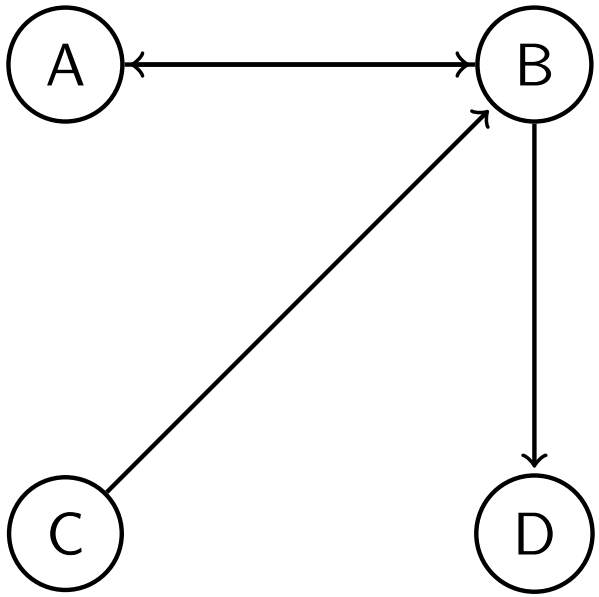
In this graph, the degree of vertex A is 3 because it has 3 edges which are incident with it.

Graph terminology

- A graph G is a subgraph of a graph H if the nodes of G are a subset of the nodes of H **and** the edges of G are the edges of H on the corresponding nodes.
- A path in a graph is a sequence of nodes connected by edges.
- A graph is connected if every two nodes have a path between them.
- A path is a cycle if it starts and ends with the same node.
- A simple cycle is one that contains at least 3 nodes and repeats only the first and last nodes.
- A graph is a tree if it is connected and it contains no simple cycles.

Directed graphs

Directed graphs are graphs in which every edge has an orientation/direction



Here $V = \{A, B, C, D\}$ and $E = \{(A, B), (B, A), (C, B), (B, D)\}$.

Strings and Languages

- An alphabet is non-empty finite set. The members of the alphabet are the symbols. An alphabet is usually written as an upper-case Greek letter. e.g. $\Gamma = \{0, 1, 2\}$.
- A string over an alphabet is a finite sequence of symbols from the alphabet. e.g. if $\Sigma = \{0, 1\}$ then 010011 is a string over Σ.
- If w is a string over Γ, the length of w, written $|w|$ is the number of symbols it contains. E.g. $|010011| = 6$.
- The string of length zero is written as ϵ and is called the empty string.
- If $w = w_1 w_2 \cdots w_n$ is a string over Γ, then $w^{rev} = w_n w_{n-1} \cdots w_2 w_1$ is the string reversed.

Strings and Languages

- If $x = x_1 \cdots x_n$ is a string of length n and $y = y_1 \cdots y_m$ is a string of length m then the concatenation of x and y is the string

$$xy = x_1 x_2 \cdots x_n y_1 y_2 \cdots y_m.$$

- The concatenation of a string x with itself several times is written

$$\overbrace{xx \cdots x}^{k} = x^k.$$

- A language is a set of strings.

- If A is an alphabet, then A^* is the set of all strings (including the empty one) over A. For example,

$$\{0,1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots\}.$$