Computation, Encoding and Languages

- Computational Problems, Strings and Data Encoding
- Binary Encoding
- Language
- Versions of Computational Problems
- Decision Problems as Language Recognition
- Models of Computation – CPU + Memory

Imdad ullah Khan

Versions of Computational Problems

What is a computational problem?

A computational problem is characterized by three things

- $\mathcal{I}$: set of (valid) input instances
- S: solution space, set of possible solutions for instances in $\mathcal{I}$
- $f : \mathcal{I} \to S$: The computational question or function

Instance $\in \mathcal{I}$	Solution $\in S$
0	No
1	No
2	Yes
3	Yes
4	No
5	Yes
6	No
7	Yes
$\vdots$	$\vdots$

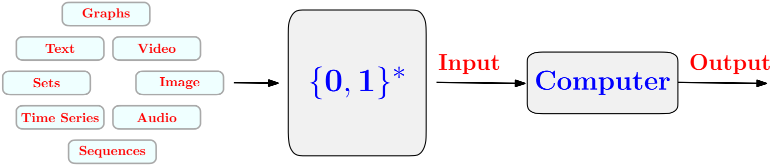
Instance $\in \mathcal{I}$	Solution $\in S$
$0, 0$	0
$0, 1$	0
$1, 2$	2
$2, 3$	6
$2, 7$	14
$2, 5$	10
$4, 6$	24
$9, 7$	63
$\vdots$	$\vdots$

Instance $\in \mathcal{I}$	Solution $\in S$
$[0, 3, 2, 9, 5]$	$[0, 2, 3, 5, 9]$
$[9, 8, 6, 4, 3, 1]$	$[1, 3, 4, 6, 8, 9]$
$[1, 2, .3, 8, 7.4]$	$[1, 2, .3, 7.4, 8]$
$[-1, 2, 9, 7, 6]$	$[-1, 2, 6, 7, 9]$
$[4, 5, 6, 7, 8]$	$[4, 5, 6, 7, 8]$
$[-2.3, -5.9, -.4.3]$	$[-5.9, -.4.3, -2.3]$
$[7, -3.6, 9, .8, .5]$	$[-3.6, .5, .8, 7, 9]$
$\vdots$	$\vdots$

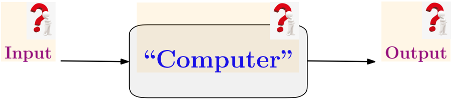
What is a computational problem?

A computational problem is characterized by three things

- $\mathcal{I}$: set of (valid) input instances
- S: solution space, set of possible solutions for instances in $\mathcal{I}$
- $f : \mathcal{I} \rightarrow S$: The computational question or function

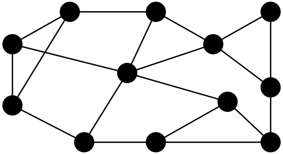


Next we talk about the output

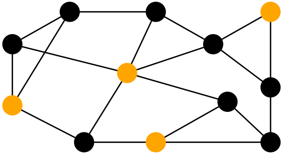


Independent Set in Graphs

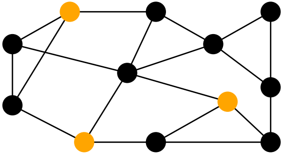
An independent set in G is subset of vertices no two of which are adjacent



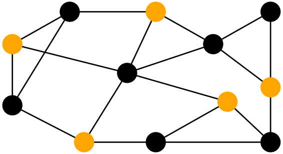
A graph on 12 vertices



An independent set of size 4



An independent set of size 3



An independent set of size 5 (max)

The IND-SET(G, k) problem: Is there an independent set of size k in G?

Independent Set Applications

Sites Selection Problem

Suppose n potential sites are identified for opening k restaurants

Some pairs of places shouldn't have the franchises at both of them

▷ too close to each other, competitions, or operational constraints

1. Make a graph G with vertices as sites and edges as pairwise conflicts
2. Selecting k sites becomes finding a k-independent set in G

Independent Set Applications

The SNP (Single Nucleotide Polymorphism) Assembly Problem

In computational biology (biochemistry) given a set of sequences we want to resolve inter-sequential conflicts by excluding some sequences

▷ conflict between sequences is due to their biochemical properties

The goal is to select a large number of conflict free sequences

1. Make a graph with vertices representing sequences and edges representing conflicts
2. Find a large independent set in this graph

Independent Set Applications

Diversifying Investment Portfolio

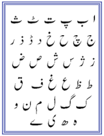
- Different stocks in a market
- $P_i(t)$ is price for stock i at time t
- $R_i(t) = \log \dfrac{P_i(t)}{P_i(t-1)}$, return or trading volume of stock i at time t
- Make each stock a node and two stocks have edges if correlation of their returns is $\geq \theta$ for threshold $-1 \leq \theta \leq 1$
- θ is set depending on potential risk (degree of diversification)
- Two adjacent vertices in $G_{\theta=.9}$ represent high risk investment pair

Set $\theta < -0.5$: an independent set in G_θ represents a portfolio with "small" risk (diverse set of investments)

Independent Set Applications

Shannon Capacity of a graph

Sending a message from an alphabet through a noisy channel



Because of noise some characters can be confused

How many 1 length strings can be sent without confusion?

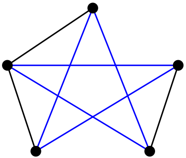
- Make each letter a node and make edges iff the corresponding letters can be confused (depends on the SNR of channel)
- Max number of messages is the size of max independent set

How many k-length strings can be sent on this channel?

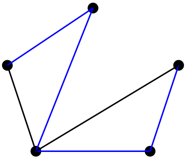
- Size of max independent set in G^k (strong product of graphs)

(Directed) Hamiltonian Cycle and Path

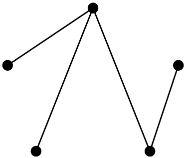
A Hamiltonian cycle (path) in graph is a cycle/path containing all vertices



Hamiltonian cycle in G



No Hamiltonian cycle in G
Hamiltonian path in blue



No Hamiltonian path in G
So no hamiltonian cycle

HAM-CYCLE(G) problem: Does G have a Hamiltonian cycle?

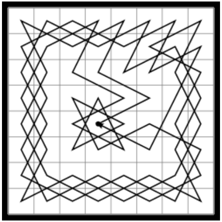
HAM-PATH(G) problem: Does G have a Hamiltonian path?

DIR-HAM-CYCLE(G), and DIR-HAM-PATH(G) are defined analogously

Hamiltonian Cycle and Path Applications

Is there a sequence of moves that takes the knight to each square on an 8×8 chessboard exactly once, returning to the original square?

For 8×8 Abu Bakr Muhammad b. Yahya al-Suli found one in 9th century



1. For $n \times n$ chessboard define a vertex for each position and connect vertex v_{ij} to vertex v_{kl} if there is a legal move between the (i, j)th position to the (k, l)th position on the board
2. Find a Hamiltonian cycle in the graph

Hamiltonian Cycle and Path Applications

Route for School Bus

School bus should visit each house exactly once to save fuel and time

1. Houses considered nodes and streets as edges
2. Find a Hamiltonian cycle

Hamiltonian Cycle and Path Applications

Genome Mapping

Combine many tiny fragments of genetic codes (called "reads"), into one genomic sequence

1. Consider each read a node in a graph
2. Overlap (end of one read matches the start of another) is an edge
3. Find a Hamiltonian cycle in this graph, a mapping of genome

The Satisfiability Problem : SAT

- Given n Boolean variables $x_1, \dots, x_n$
- A literal is a variable appearing in some formula as x_i or $\overline{x_i}$
- A clause is an OR of one or more literals
- A CNF formula (conjunctive normal form) is a Boolean expression that is AND of one or more clauses
- A formula is satisfiable if there is an assignment of $0/1$ values to the variables such that the formula evaluates to 1 (or true)

1. $f_1 = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x_2}) \wedge (x_2 \vee \bar{x_3})$
    - f_1 is satisfiable (the assignment is $x_1 = 1, x_2 = 1, x_3 = 1$)
    - $x_1 = 1, x_2 = 0, x_3 = 0$ is also a satisfying assignment
2. $f_2 = (x_1 \vee \bar{x_2}) \wedge (x_1 \vee x_2) \wedge (\bar{x_1} \vee \bar{x_2}) \wedge (\bar{x_1} \vee x_2)$ is not satisfiable

The SAT(f) problem: Is there a satisfying assignment for the formula f?

The Satisfiability Problem : 3-SAT

- Given n Boolean variables $x_1, \ldots, x_n$
- Each can take a value of 0/1 (true/false)
- A literal is a variable appearing in some formula as x_i or $\bar{x}_i$
- A clause of size 3 is an OR of three literals
- A 3-CNF formula is AND of one or more clauses of size ≤ 3
- A formula is satisfiable if there is an assignment of 0/1 values to the variables such that the formula evaluates to 1 (or true)

The 3-SAT(f) problem: Is there a satisfying assignment for f?

The Satisfiability Problem :Applications

Many applications in hardware/software verification, Also in planning, partitioning, scheduling, constrained satisfaction problem

Many hard problems can be stated in terms of SAT

When can the meeting take place if at all, with the following constraints?

- John can only meet either on Monday, Wednesday or Thursday
- Catherine cannot meet on Wednesday
- Anne cannot meet on Friday
- Peter cannot meet neither on Tuesday nor on Thursday

Encode them into the following Boolean formula:

$$(Mon \vee Wed \vee Thu) \wedge (\neg Wed) \wedge (\neg Fri) \wedge (\neg Tue \vee \neg Thu)$$

The meeting must take place on Monday

Versions of Computational Problems

Decision Problem ▷ Sometimes called decision version of a problem

Characterized by their algorithms whose output is either **Yes** or **No**

In other words the answer on an instance is either **Yes** or **No**

- Decision versions of SAT(f) asks if the given formula f is satisfiable
  - output is **Yes** if there is an assignment to variables that makes f true
- Decision version of IND-SET(G, k) asks if there is an independent set of size k in G
  - output is **Yes** if G has an independent set of size $\geq k$, else **No**
- Decision version of HAMILTONIAN-CYCLE(G)
  - Output is a **Yes** if G has a Hamiltonian cycle, else **No**

Versions of Computational Problems

Search Problem ▷ Sometimes called search version of a problem

Ask for a structure satisfying certain property or NOT-FOUND= **NF** flag

The expected answer on an instance is not (necessarily) **Yes** or **No**

- Search versions of SAT, 3-SAT ask for a satisfying assignment
  - output is n-bit string (specifying ordered values for variables) or **NF**
- Search version of IND-SET(G, k) asks for an ind. set of size k in G
  - output is a subset of vertices or **NF**
- Search version of HAM-CYCLE(G)
  - output is a Hamiltonian cycle in G or **NF**

Versions of Computational Problems

Optimization Problem ▷ also called optimization version of problem

These problems ask for a structure that satisfy certain property (feasibility) and no other feasible structure have better **value**

These are search problem but searching for an optimal structure

▷ There is an objective/value function over solution space

- Optimization versions of SAT, 3-SAT ask for an assignment satisfying the most number of clauses
  - output is n-bit string (specifying ordered values for variables)
- Optimization version of IND-SET(G) asks for largest indep. set in G
- What could be optimization version of HAM-CYCLE(G)?

▷ TSP(G) asks for a minimum cost TSP tour

Versions of Computational Problems

- Decision Problem: answer is **Yes/No**
- Search Problem: answer is a feasible structure of certain value or **NF**
- Optimization Problem: answer is a feasible structure of optimal value

In some cases there is no reasonable notion of optimization version e.g. Hamiltonian cycle