# Principle of Information & Security

Siddharth Bhat

# Contents

**1 Problems, Solutions, and Resources** **5**

1.1 Problems . . . 5

1.1.1 Kannan . . . 6

1.2 Solutions . . . 6

1.2.1 Kannan . . . 6

1.3 Resources . . . 6

**2 Diagonalization** **7**

2.1 Hierarchy theorems . . . 9

2.1.1 Proof sketch . . . 9

# Chapter 1

# Problems, Solutions, and Resources

## 1.1 Problems

Alphabet set is finite, call it $\Sigma$. Strings must be finite length.

Given some input, and a computer that produces some output, the description could be infinite — both input and output.

However, the machine's *description* (aka, the relationship between input and output) must be finite.

So, the *total input* can be infinite, but the input chunk must be finite, and the response of the machine per *input chunk* must be finite.

So, we can just use the language $L = \{0, 1\}$ for the machine.

Problems which have yes/no as answers are called decision problems. Inputs are from $\Sigma^*$, outputs are from $\{0, 1\}$. The problem is a mapping $f : \Sigma^* \to \{0, 1\}$. This is equivalent to providing the set $\texttt{ACCEPT} \subset \Sigma^* = f^{-1}(1)$. Note that $\texttt{REJECT} = \texttt{ACCEPT}^c$. The set `ACCEPT` is called a language.

Now, we can study languages by looking at their grammars (welcome, Chomsky).

What about fractional bit problems? Is this useful? Could we exploit some properties of fractional dimension?

### Cantor set

take $S_0 = [0, 1]$ In each iteration, remove the middle one-third of each continuous interval. Therefore,

- $S_0 = [0, 1]$
- $S_1 = [0, \frac{1}{3}] \cup [\frac{2}{3}, 1]$

In $S_\infty$, *uncountably infinite* points remain (However, this set has *measure* 0).

So now, the question is, what is the dimension? We define Haussdorf dimension, and use this to exhibit fractional dimension of the Cantor set.

`TODO: fill this up!`

The total number of problems that can exist is $powerset(\Sigma^*)$. `RE` (recursively enumerable) Is a subset of $powerset(\Sigma^*)$ which computers can handle. The annoying thing is that there are *finite length problems* which computers cannot solve.

### 1.1.1 Kannan

If the universe is a machine, then it must have infinite description.

QM is the meeting point of universes?

## 1.2 Solutions

### 1.2.1 Kannan

**Question:** We study a lot of Science — why? What is the ultimate goal of science? Equivalently, what is the theory of everything we need to find to halt on the journey of Science?

Assuming Science = God, we need to ask Science a question. Which language will we use to query Science? Or, well, which language is *enough* to query Science? If the query alphabet is $\Sigma$, we can ask $\Sigma^*$ questions. However, we can only reasonably pose questions of finite length (even though the Science oracle can answer questions of infinite length).

In this case, have we achieved te ultimate goal of science?

## 1.3 Resources

## Chapter 2

# Diagonalization

- Level 1: $\mathbb{R}$ is uncountable.
- Level 2: $\exists L, L \notin \texttt{RE}$.
- Level 3: Halting problem is undecidable.
- Level 4: Time/Space hierarchy.
- Limitations: Exists oracles A, B such that — $P^A = NP^A, P^B \neq NP^B$
- Level 5: If $P \neq NP$, $\exists L, L \notin P, L \notin NPC$ (Ladner's theorem)

Diagonalization cannot separate P, NP — If it could, then it should also separate P with any oracle, and NP with the same oracle. We know that there exists an oracle such that we can separate $P^A = NP^A$, as well as $P^B \neq NP^B$.

chapterReview of the last 3 lectures, after add-drop

- Is it easier to *pose* problems than to *solve* them?
- Can every "solvable" problem have a solution that uses finite resources?
- What problems are *interesting*?
- Are all interesting problems solved in an *interesting* way? (P v/s NP)
- Can things get more interesting? (Quantum Mechanics, Approximation, Randomness, Interactivitiy, . . . )

Are there problems with infinite length input / output but can still be posed in finite time? Eg. output $\pi$ in decimal. However, we decided that both input/output should be finite. We decided this does not belong to problems we wish to solve it, since we cannot solve it in finite time. If we believe that nature is inherently noisy, or nature is quantized, or nature has finite precision, then we cannot consider problems that require infinite time as problems in this universe (since Nature / the universe itself cannot pose such a problem).

Quantum mechanics (which is a theory of quantization) is developed over infinite precision mathematics ($\mathbb{C}$). Does this really make sense? There is a way in which a quantized universe can be infinite precision: This is by using 'external help': There are infinite such quantized universes which intersect at some points, and at those points, precision will increase. (If we both have a resolution of 1 pixel but are at a gap of 1/2, my least count is now 1/2). If there are an infinite number of universes overlapping at a single point, then we can construct "infinite precision". (*I feel this is crazy. Is this really crazy?*)

Posing a question is creating a language $L \subset \Sigma^*$. (Sid: a solution is a classifier for $L$).

Kannan's view:

- Finite space $\equiv$ finite information can be stored. (Turing: finite tape alphabet. Since a cell demarcates a finite volume, we want to have a finite amount of info in this cell)
- Information travels at finite speed. If we have cells, we should not be able to store and retreive information "equally" (based on how far we are from it). Hence, all infinite memory must be sequential memory since information travels at finite speed.
- Finite program $\equiv$ finite control.

Solution to these choices is a TM.

Do all languages have a TM recognizing it? No ($\texttt{RE}$ = solvable by TM).

The class $\texttt{R}$ = decidable by a TM (TM halts on all inputs). Diagonalization led us to Halting problem.

We have the class $P$, and we claimed that $P$ is interesting. Given that $P$ is considered interesting because of feasibility, it is possible that there are questions that are interesting even though **solving them** is not feasible. For example, if we can actually **understand** the solution, or the proof of non-existence of solutions, then we will care. `IP = PSPACE` is one such magical case where if someone can solve with a lot more power than you have access to, you can learn things from them interactively in reasonable time.

## 2.1 Hierarchy theorems

$\exists L$, such that $\forall f : \mathbb{N} \to \mathbb{N}$, where $f$ is space/time constructible,

$$Space(f) \supsetneq Space(o(f))$$
$$Time(f) \supsetneq Time\left(\frac{o(f)}{log f}\right)$$

So, there is a Hierarchy of complexity classes in time and space.

### 2.1.1 Proof sketch

We exhibit a languag $A$, such that $A \in Space(f(n))$, and $A \notin Space(o(f(n))$.

Let $D$ decide $A$. D's definition:

- compute $f(n)$ and mark the end of $f(n)$ cells. If the read-write head ever crosses it, `REJECT`, `HALT`. We first need $f(n)$ to use $f(n)$ cells or less to compute. This is called as **space-constructribility**. ($f : \mathbb{N} \to \mathbb{N}$ is space-constructible iff given $n$, $\exists$ TM which computes $f(n)$ using at most $f(n)$ cells). Also, we want $f(n)$ to be at least $log(n)$. Clearly, this process is in space $f(n)$.
- We now need to "seaparate" A from the smaller classes. If A can be solved in a smaller space (ie, we cannot separate $A$), then there must be a TM (say, $D'$) which decides A in space less than $f(n)$. So now, we need to choose some input such that $D'$ is different from $D$. We can use diagonalization to construct such a function.
- let the input be $x$. Let $x = M10*$ for some TM $M$. if not, `REJECT`, `HALT`.
- Let $D$ simulate $M$ on input $M$. If $M$ takes less that $f(n)$ time to run on $M$, then $M$ can decide A in time less than $f(n)$. So now, $D$ knows how much space $M(M)$ requires. if $M(M)$ accepts, we reject. If $M(M)$ rejects, we accept (diagonalization).
- To find out whether $M(M)$ rejects, note that it is space-bounded, so we can just check how many states of the configuration space it visits. If it has not halted after visiting all states in the configuration space, we can conclude that $M(M)$ does not halt. The configuration space is $O(2^{f(n)})$.

**Arjun Q:** Are there examples of non-space constructible functions, which are non-trivial? Other than ones that are too-small?