Abstract

An overview of algorithms, words and definitions from my studies as a computer scientist BSc, with an interest in algorithms.

Note that this document is not intended to be a *public friendly* document; it is written by one author and for one author, only. But you are still free to read.

Contents

List of Figures

List of Tables

Chapter 1

Words and Definitions

1.1 Algorithms

Definition (Approximation algorithm). Find approximate solutions to Optimization problem

Definition (Asymptotic Polynomial-time Approximation Scheme, APTAS). A family of algorithms, and a constant c such that all solutions has an approximation of $(1+\epsilon) \times OPT + c$ for minimization problems.

Definition (Bin-packing). A series of algorithms to learn how to distribute n numbers into k bins. First-fit, best-fit, worst-fit (stack into where there is most free space), best-fit, etc.

Definition (Complimentary slackness). Given an optimal solution to a linear program, Z_{LP} and it's dual Y_{LP}, with $x_1, x_2, \dots x_n$ and $y_1, y_2, \dots y_n$ respecively, with $w_1, w_2, \dots w_n$ and $z_1, z_2, \dots z_n$ as slack variables for each solution, respectively, then $\forall x, x_i z_i = 0$ and $\forall y, y_i w_i = 0$

This necessary condition for optimality conveys a fairly simple economic principle. In standard form (when maximizing), if there is slack in a constrained primal resource (i.e., there are "leftovers"), then additional quantities of that resource must have no value.

Definition (Difference heuristic and approximation). While an heurisitc makes a choice, without a guarantee of optimality, an approximation can make a choice and know that this choice will render a solution within a factor of OPT.

Definition (EDD). Earliest Due Date

Definition (F-approximation). Also referred to as a linear approximation, using a function f, which is affine.

Definition (FPTAS, fully polynomial approximation scheme). As in PTAS, Polynomial-time approximation scheme, just $\frac{1}{\epsilon}$. The algoritm is required to be polynomial both in running time and problem size.

Note that strongly NP-complete problems do not have any FPTAS.

Definition (Integrality gap). The biggest difference between an IP and LP

Definition (Makespan). The total length of a schedule; from 0 to C_{max}

Definition ($\tilde{O}$). Given function $f(x)$, $\tilde{O}(f(x)) = O(f(x) \cdot \log^k f(x)$

Definition (Optimization problem). To find the best solution of n feasible solutions.

Definition (Perfect matching). A collection $E' \subseteq E$ of edges in a graph $G = (V, E)$, such that $\forall v \in V$, are connected from E' only once.

Definition (Pre-empty schedule). You can interrupt task and re-continue them.

Definition (PTAS, Polynomial-time approximation scheme). Given an optimization problem (e.g. an NP-problem) and a parameter ϵ, produce a solution within $((1+\epsilon) \times OPT)$ $\epsilon > 0$

E.g. for the traveling salesman, a tour would be of length max $(1+\epsilon) \times L$, with L being the length of the tour

Note that for minimization, there is $1+\epsilon$, and for maximization, there is $1-\epsilon$

If you have a scheme with $(1 \pm \epsilon) \times OPT + \kappa$, then it is not under PTAS. PTAS only handles the former part, $(1 \pm \epsilon)$

For MAX SNP, there does not exist polynomial approximation schemes

Definition (ρ-approximation). Polynomial algorithm that is guaranteed to have objective function to OPT within ρ of optimum (not the (1+ϵ) of PTAS, Polynomial-time approximation scheme).

Definition (Scheduling). See Longest processing time rule,

- P_i = time to do a job i
- R_i = earliest time a job i can start
- C_i = time of completion for job i
- D_i = due date for job i
- $L_i = C_i - D_i$

Definition (Strong duality). The optimal value of the dual is equal to that of the primal linear program.

$\sum y_i^* = \sum w_i x_i$

Definition (Weak duality property). No dual program has a solution greater than the optimal of the primal linear program

Definition (α approximation). Produce a solution who's value is within a factor of α of the optimal.

1.2 Calculus

Definition (Affine). $f(x_1, x_2, \dots, x_n) = a_1x_1, a_2x_2, \dots a_nx_n$

Definition (arcsin).

$$\sin y = x$$
$$\arcsin x = \sin^{-1} x = y$$

Properties:

- $\arcsin x = \frac{\pi}{2} - \arccos x = 90 - \arccos x$
- $\cos(\arcsin x = \sin(\arccos x) = \sqrt{1 - x^2}$
- $x = -1 \rightarrow \arcsin x = -1 \times \frac{\pi}{2}$
- $x = 1 \rightarrow \arcsin x = \frac{\pi}{2}$
- $x = 0 \rightarrow \arcsin x = 0$

Definition (arccos). Properties:

- $x = -1 \rightarrow \arccos x = \pi$
- $x = 1 \rightarrow \arccos x = 0$
- $x = 0 \rightarrow \arccos x = \frac{\pi}{2}$

Definition (Arc length). Length of a curve when straightened out.

Definition (arccos). While cosine shows the relation between lengths in a triangle, arccos gives the angle.

Definition (Arithmetic-geometric mean inequality).
$(\prod\limits_{i=1}^{k} a_i)^{1/k} \leq \frac{1}{k} \sum\limits_{i=1}^{k} a_i$

Definition (Bijection).

$$S, R \text{ are sets} \tag{1.1}$$
$$\forall i \in S, \exists! f(i) \in R \wedge \tag{1.2}$$
$$\forall i \in R, \exists! f(i) \in S \tag{1.3}$$
$$\tag{1.4}$$

Definition (Convolution). Dictionary: a coil or twist, especially one of many.
Informal: an expression of how a shape from one function is modified by the other.

Can also be used to smoothen a discontinous a function (making it continous on the given range). We need to normalize our g(x - τ) so that we do not continously increase the f(x)

Definition (Continuous variables)**.** There are three types of continuous variables:

Nominal Categorized within groups: box 1, 2, 3 or 4.

Dichotomous Boolean, yes or no.

Ordinal A nominal variable, just that different groups give different values. E.g. to like something on a scale of 1 to 10 gives you a ordinal range

Definition (Concave)**.** Let f be a function defined on the interval $[x_1, x_2]$. This function is concave according to the definition if, for every pair of numbers a and b with $x_1 \leq a \leq x_2$ and $x_1 \leq b \leq x_2$, the line segment from $(a, f(a))$ to $(b, f(b))$ lies on or below the function.

- The sine function is concave on the interval $[0, \pi]$
- Concave if every line segment joining two point is never above the graph
- Concave functions has $f''(x) \leq 0$ in a given interval

Definition (Differential operator)**.** An operator to do differentiation. This is mainly to abstract differenentiation.

Definition (Divergence)**.** measures the magnitude of a Vector Field's source or sink at a given point, in terms of a signed scalar.

E.g. air can be thought of to have a point s and t, where they push out hot and cool air, respectively. From these points, you can create a Vector Field that shows how air spreads from s and t. The divergence measures the collected value from each of these Vector Fields.

The operator for divergence is `{div}`.

Given vector field $F = Ui, Vj, Wk$:

$$div\mathsf{F} = \nabla \cdot F = \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z} \tag{1.5}$$

Definition (Elementary function)**.** In mathematics, an elementary function is a function of one variable built from a finite number of exponentials, logarithms, constants, and nth roots through composition and combinations using the four elementary operations $(+)$.

Definition (Extreme point)**.** A point furthest away from something.

Definition (Field)**.** A physcial quantity that has a value for each point in space and time.

Definition (Flow)**.** Motion of particles in a given set.

Definition (Generalized function)**.** A distribution without steps, i.e. it is continious.

Definition (Harmonic numbers)**.** $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + 1n = \sum_{k=1}^{k} \frac{1}{k} \simeq \ln n$

Definition (Heavyside Step Function).

$$H(x) = \begin{cases} 0 & \text{for} x < 0 \\ \frac{1}{2} & x = 0 \\ 1 & \text{for} x > 0 \end{cases}$$

Definition (Identity function). $\forall x, f(x) = x$

Definition (Integral). The "reverse" to a derivate.

$$\int_a^b x^n dx = \frac{x^{n+1}}{n+1} + C \tag{1.6}$$

$$\tag{1.7}$$

Definition (Law of cosines).

$$v \cdot u = |v||u| \cos \theta$$

Note that if v, u are unit vectors, their lengths are both 1. We can then rewrite the expression as

$$\begin{aligned} v \cdot u &= \cos \theta \\ \arccos (v \cdot u) &= \theta \end{aligned}$$

Definition (Laplace operator). Transform a function of f of t to a function f of s

Definition (Laplacian Matrix). sometimes called admittance matrix, Kirchhoff matrix or discrete Laplacian, is a matrix representation of a graph.

Definition (Line integral). An integral where the function to be integrated is along a curve. The function is usually a Vector Field or Scalar field.

Definition (Partial derivative). Given a function f with multiple parameters, a partial derivative is a derivative with respect to one of those variables.

Partial derivatives are often denoted by ∂.

To use partial derivation, you often assume that the other variables are constants. Otherwise, there is an infinite number of tangent lines at any point, so you will have to have some range on the tangents.

Definition (Parameterization). Represent a curve as a function.

$$x = \cos t \tag{1.8}$$

$$y = \sin t \tag{1.9}$$

is the parametric representation of a unit circle.

Definition (Partial differential equation). A set of variables, and equations that show how they are all linked together.

Definition (Rectangle function)**.**

$$\Pi(x) = \begin{cases} 0 & \text{for} x > \frac{1}{2} \\ \frac{1}{2} & \text{for} x = \frac{1}{2} \\ 1 & \text{for} x < \frac{1}{2} \end{cases}$$

Definition (Riemanns sum)**.** Sum from an integral. Divide the area under/over a curve into rectangles or trapezoids, and sum together their area. The smaller the shapes, the better.

Definition (Standard form)**.** To write a number as a power of 10.

Definition (Vector Field)**.** An assigment of direction for a given set of points in an euclidian space. E.g. select every (10n, 10n) pixels in an image and get their derivative.

1.3 Graphs

Definition (Arborescence). an arborescence is a directed graph in which, for a vertex u called the root and any other vertex v, there is exactly one directed path from u to v.

Definition (Clique). A subset of vertices $C \subset V$, such that in this subgraph all nodes are connected, i.e. there is an edge from every pair of nodes.

Definition (Connected graph). A graph in which, from any $v \in V$, you can reach any other $u \neq v$

Definition (Directed graph). Properties:

- $\sum\limits_{v \in V} d(v) = 2|E|$
- $\sum\limits_{v \in V} indegree(v) = \sum\limits_{v \in V} outdegree(v)$

Definition (Eularian). Visit each *edge* once.

Definition (Hamiltonian). Visit each *vertex* once.

Definition (Metric spaces). See also Semi-metric Properties:

- $d_{u,v} = 0 \iff u = v$
- $d_{u,v} = d_{v,u}$
- $\forall k, d_{u,v} \leq d_{u,k} + d_{k,v}$

Definition (Minimum mean cost). minimize ratio of cost of arcs (directed edges) to number of arcs

Definition (MST). A mininum Spanning tree, such the weight of this tree is less than or equal to all other possible Spanning tree.

Definition (Planar graph). No edges need to cross each other.

Definition (Semi-metric). Like a Metric spaces, without the property that $d_{u,v} = 0 \iff u = v$

Definition (Spanning tree). Connected, undriected graph that has all vertices and some subset of edges to form a tree.

Definition (Topological sort). Runs in O(V + E), same as DFS, although it can be supered to $O(\log_2 n)$

Definition (Tree). Properties of a tree:

- in a tree, there will always be an even amount of nodes that has an uneven degree.
- Degree of nodes in a tree is at most twice the number of nodes

Definition (Vertex-cover). A selection of vertices such that each edge is incident to at least one of them.

1.4 Image Processing

Definition (Anisotropic diffusion). also called Perona-Malik diffusion, is a technique aiming at reducing image noise without removing significant parts of the image content, typically edges, lines or other details that are important for the interpretation of the image.

Definition (extrapolation). The process of estimating, beyond the original observation range, the value of a variable on the basis of its relationship with another variable.

Creating a tangent line at the end of the known data and extending it beyond that limit.

Definition (Gaussian blur). Use a gaussian function on an image to reduce image noise and detail

Definition (Gaussian filter). Gaussian filters have the properties of having no overshoot to a step function input while minimizing the rise and fall time.

Definition (Gradient flow).

$$V = \nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)$$

Definition (Image gradient). Gradual blend of color

Definition (Image noise). Unwanted signal, electrical signals not wanted in an image

Definition (Poisson image editing). Blend two images together and make them seem alike.

Definition (Raster order). Begin at top left, proceed to right, then at leftmost pixel in next line.

Definition (Scalar field). Associate a value to every point in a space. E.g. for an image, you could assign each pixel a color value.

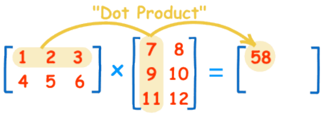
1.5 Linear Algebra

Definition (Basis). Given a set of vectors in $\mathbf{R}^n, V$, which is linearly independent, the set V is a *basis* if you can span $\mathbf{R}^n$ using V.

Definition (Column space). All linear combinations of the columns of a matrix A.

Definition (Diagonal Dominance). $\forall i, \exists i A_{i,i}, \forall i, \iff A_{i,i} \geq \sum_{j=0, j\neq i}^{m}$, then matrix A is DD.

Definition (Dotting matrices).



Definition (Eigenvector). Given a square matrix A, when A is multiplied with an eigenvector v, the resulting matrix A′ is a multiple of v. The multipe is denoted by λ and is called an eigenvalue. So, $Av = \lambda v$

Definition (Gaussian Elimination). AKA "Row reduction". Add row x to row $y, y \neq x$, to reduce y to zeroes.

Definition (Inverse). For a matrix $A \in \mathbf{R}^{2x2}$: $A^{-1} = \frac{1}{|A|}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

For a matrix $A \in \mathbf{R}^{3x3}$: bcacab

Properties:

- $(A^{-1})^{-1} = A$

Definition (Kernel). For a vector space given by a Linear transformation, a kernel is the set of vectors $v \in V$, s.t. $T(u) = 0$.

Definition (Length of Vector). For a vector

$$v = [a_1, a_2, \dots, a_n]$$
$$|v| = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2}$$

Definition (Linear dependence). For $V = v_1, \dots, v_n$, and $\forall v, visvector$, if none of the vectors in V can be written as a linear combination from the other vectors in V, the set is linearly independent.

Definition (Linear transformation). Take a vector space into another, s.t.

$$T(u+v) = T(u) + T(v) \forall u, v \in v \quad (1.10)$$
$$T(cu) = cT(u) \forall u \in V \quad (1.11)$$

Definition (Normalization of vectors). $\hat{X} \equiv \frac{X}{|X|}$, where $|x|$ is the Length of Vector X is, in this case, evaluated as the additive sum of it's entries.

Definition (Null space)**.** Given a matrix A, if you solve for that each row = 0, all possible values for each x makes out the Null space.

Note that here it is apossible to get free variables for some x, and bound to others.

Definition (Orthogonal)**.** Two lines that intersect each other at 90 degrees.

- Orthogonal matrices preserve dot products: given two vectors u, and v, and an orthogonal matrix Q, the following is true: $u \times v = Qu \times Qv$
- The determinant of an orthogonal matrix is always 1 or -1
- The transpose of Q is equal to it's inverse, hence: $Q \times Q^T = I$

Definition (Perpendicular)**.** Similar to Orthogonal, but with lines.

Definition (Plane)**.** A flat, two-dimensional surface

Definition (Positive semidefninite matrix)**.** Properties:

- Nonnegative Eigenvectors
- $X = V^T V$ for some $V \in \mathbf{R}^{mxn}$
- $X = \sum_{i=1}^{m} \lambda_i w_i w_i^t$ for some $\lambda_i \geq 0$ and vectors $w_i \in \mathbf{R}^n$ such that $w_i^T w = 1$ and $w_i^T w_j = 0$

Definition (Projection)**.** define a vector v and u.

$$\begin{aligned} L &= \{cv | c \in \mathbf{R}\} \\ proj(v) &= l \in L \text{ such that } u - proj(v) \text{ is Orthogonal to l} \\ \textit{I.e.}, proj(v) &= cv, c \in \mathbf{R} \end{aligned}$$

Properties:

- $proj(v) = proj(v)^2$
- Linear independence on u, v also relates for $v - proj(v), u$
- adding $proj(v)$ to v gives you u

Definition (Span of vectors)**.** All linear combinations of a set of vectors.

$$\begin{aligned} V &= \{v_1, \ldots, v_n\} \\ C &= \{c \in C | \mathbf{R}\} \\ span &= c_1 v_1 + \cdots + c_n v_n \end{aligned}$$

Definition (Symmetric)**.**
- Length of rows is equal
- The transpose is equal to the originl

Definition (Tensor)**.** Geometric objects that describe linear relations between vectors or scalars.

Definition (Trace)**.** Sum of all diagonal entries in a matrix. $A_{11} + A_{22} + A_{nn}$

Definition (Transpose)**.** Take column i and make it into a column. Repeat.

Definition (Unit vector)**.** A vector who's length is 1.

Definition (vector length)**.** number of "steps" in a vector

1.6 Programming Security

Definition (Accessory controls)**.** If a company X notices that a part P of a product does not belong to them, they may reduce the effectiveness of the product.

For example, if a printer notices a competitors ink in the printer, it may go from high quality to low.

This is considered a form of authentication.

Definition (Access control)**.** I assume the meaning is intuitive to understand. What's worth mentioning is that access control can operate on four levels, namely:

- Application
- Middleware
- Operating system
- Hardware

Definition (Access Control Lists)**.** Most UNIX-fans will know them already: *-rw-r–r– 1 <owner of file> <group owner of file>.* This is an example of a rowwise mandatory ACL- where an object has a given set of definitions for three different groups: the first three letters in the string denotes what the owner can do (in this case, there is no execution, but read and write), the third next letters (mid-block) denote what members of the owner group can do to the file, and the next three letters denote what everyone else can do.

This type of ACL are slow for systems with many users. The vantage is that is efficient. Other forms of ACL's include database access control (suffering from inability to model states, etc.

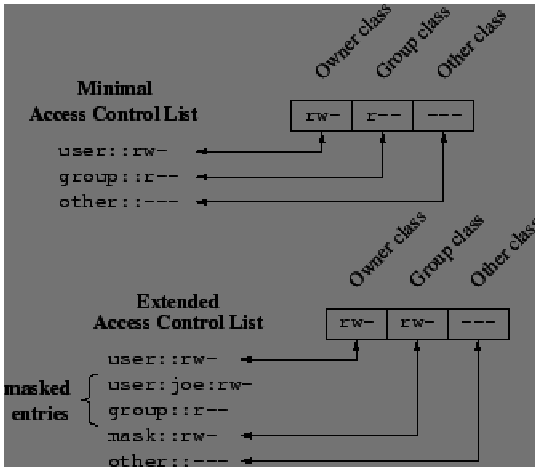


Figure 1.1: Typical UNIX ACL

Definition (Assurance classes)**.**

Definition (Authentication)**.** To prove that you are who you claim to be. E.g. to provide a password for an account, etc.

Definition (Authorization)**.** To ensure that a user has sufficient permissions to do what he/she asks for. E.g. an authenticated user may log in, but may not change the passwords for other users, because this authenticated user is not authorized for this account.

Definition (BAN-logic)**.** IS THIS REALLY NECESSARY?
...formal reasoning in cryptology, somewhat alike first-order-logic.

Definition (Biba Model)**.** Essentially BLP, just that it is read by "no read down, no write up". The BLP can be categorized as "no write down, no read up".

A monk may write a prayer book that can be read by commoners, but not one to be read by a high priest

— Wikipedia

Definition (Bell LaPadula model)**.** Any system that adheres to the two following properties, are said to adhere to the Bell LaPadula Model, which is a Security Policy (model):

> **Simple security property** *no process may read data at a higher level.* If a process is on runlevel X, it should not have access to data at level X + 1.
>
> ***-property** *no process may write data to a lower level.* Imagine that a sysadmin gets a Trojan that reads much sensitive data. The *-property prevents this data to be written down to a lower level where it is accessible to other accounts with lower clearance.

It is worthwile noting that the properties fail to mention policies for creation and destruction of files. Furthermore, the model is broken if sensitive data is temporarily declassified. The data can then be manipulated without violating the constraints of BLP. Hence, to make the BLP safe, we need to introduce the **tranquility property**:

> **Strong:** Security labels on objects never change during systems operation.
>
> **Weak:** Security labels never change in such a way that it defines the given security policy.

However, the problem with the tranquility property is that process will have a difficult time reading files. If at first one reads something at a high level, then no concurrent read/write operation can occur for a lower level. Applications will therefore need to be customized extensively to accomodate for the tranquility property.

The BLP only deals with confidentiality, not integrity (see Biba Model)

Definition (Biba Model)**.** Deals with integrity alone, ignores confidentiality.

Definition (BMA model)**.** British Medical Association. Basically a 9-point long list wich comprehends the BLP, but also represents state, and so on. Used for medical records.

It's worthwhile noting that for medical records, it's much harder to define privacy and such because doctors are often required to involve third parties with their records. This could be researchers, drug companies, family, so on.

Definition (Capture error)**.** People are used to clicking "OK" on alert boxes without reading, so you might fool them intro clicking on one of your buttons by simply making a popup box.

Definition (Chinese Wall)**.** Lets say that you work in finance, reviewing different oil companies. If you've worked on account A for some time, you might have information that is relevant for account B. To prohibit this, you're given a timed restriction, such that you cannot work on other accounts if you have sensitive information.

The general idea is to say that "you can access this, but nothing else, there's a chinese wall between you and other accounts."

The chinese wall has been expressed to be similar to the BLP. Let c denote objects of interest, for example bank data. s denotes a subject, for example a person, who tries to access objects. $y(x)$ is a function y for an object x that denotes a company y's interest for the object x. Similarily, for a company f, there's a function $f(x)$. The domain $S = S(s)$ is a set which covers all objects that s can access.

> **Simple security property** $\forall s$, s will have access to c iff $\forall o \in S(s)$, $y(c) \cap x(o) = \emptyset$ or $y(c) \in y(o)$
>
> **The *-property** s can write to an object c iff $\forall o\{m \| m \in S(s)\}), o \notin x(c)$ and $y(c) \neq y(o)^{[what?]}$

Definition (Common Criteria)**.** Defines many Security functional requirements. ISO-standard.

Definition (Confidentiality)**.** *Who/what could have read this message?*

Definition (Covert channel)**.** If a system adheres to the BLP, a low security object might still commuicate with one at a higher level. If the two share a resource, e.g. a disc, one could make the higher level process do something with the disc head to communicate. This could for example be invoking an error at time t_i to indicate that bit i in an important file is either 1 or 0. This way of communication is called a covert channel.

Definition (Cryptographic nonce)**.** a nonce is an arbitrary number used only once in a cryptographic communication. The nonce is there to ensure freshness of a message, e.g. it could be a timestamp. That way, one can be assured that encrypted messages are recent and not old.

Definition (CSRF)**.** performs an action on the server. The user is typically not aware of what he/she is doing. I.e., if there is a URL that can be used to purchase 50 cars, Eve can send that to Alice and make her, unwillingly, buy 50 cars.

CSRF attack can be mitigated by properly implementing session authentication tokens, such that one cannot modify/use an account without having a valid token from the server. That way, the POST to purchase cannot be instantiated unless the client performs a series of steps.

Definition (Discretionary Access Control)**.** A simple form of access control. The ease of implementing DAC makes it popular. DAC is similar to Mandatory Access Control, except from that there are no *policies* for objects. This means that whenever a subject wants to apply an action to an object o, the operating system will only evaluate s and o, not the action that s wants to apply, in order to determine whether or not s's action will be executed.

Definition (Difference between DAC and MAC)**.** DAC is not a multilevel security protocol in that, for any subject s, **any** operation on an object o is either granted or not **only by considering the relationship between s and o, not the action applied**.

Furtherly, the following quotes quite sum it up:

> *Systems can be said to implement both MAC and DAC simultaneously, where DAC refers to one category of access controls that subjects can transfer among each other, and MAC refers to a second category of access controls that imposes constraints upon the first.*
>
> — Wikipedia

> *In general, when systems enforce a security policy independetly of user actions, they are described as having madatory access control, as opposed to the discretionary access control in systems like Unix where users can take their own access decision about their files*
>
> — Page 246 in the book

Definition (Evaluation Assurance Level)**.** The Common Criteria uses

Definition (Format string vulnerability)**.** Input data will be used to format output. However, the input data goes to the stack, and there are therefore vulnerabilities.

Definition (High water mark principle)**.** Start off at the bottom and elevate permissions as you need them. E.g.às one opens a mail client, one finds

Definition (Identify Friend or Foe)**.** IFF: A system that can tell whether you are a friend of a foe. A classical crypto-problem; how can I know that I am not talking to Eve?

Definition (Integer manipulation attack)**.** Causing an under- or overflow or truncation such that you can exploit software.

Definition (Integrity)**.** *Who/what could have altered this package?*

Definition (Kernel bloat)**.** E.g. in Windows, you need to have many drivers run as root in the kernel. This will naturally open up for more security holes.

Definition (Key-distribution)**.** One entity on a network wants to talk to another in a network. We will assume Alice wants to talk to Bob. Her concern is that they might establish a connection, but Eve might hijack it and pretend to be either party. Alice needs to know that she is speaking with Bob, and she needs to know that Bob is not communicating with any Eve.

To prevent this, a trusted third party is introduced. We will call this Sam. We will assume that the connection to Sam is safe, and that Sam knows every user of the network. Alice contacts Sam. Her request is as follows: "I am Alice, and want to communicate with Bob". Sam returns two tokens. The first token can only be read by Alice, and the second only by Bob. Alice then contacts Bob, giving him only the second certificate. Bob decrypts the certificate, which unravels an encryption/decryption key. Bob and Alice can now encrypt and decrypt messages to each other using this key, and communicate without Sam.

Definition (Landing pad)**.** A piece of code, such that when exectued, the processor will exectue malicious code.

Definition (Lattice model)**.** Essentially equivalent to the BLP. You use labels such as "TOP SECRET", "SECRET" and say that there are no communications between levels as in the BLP.

You combine things, so for each item, e.g. for "missiles" you might have a clearance for something like "SECRET". But you also need codewords, such as "SECRET MISSILES" if you want to read the secret stuff about missiles.

The point of the lattice model is that you have aggregated values, such that a lattice will yield a numeric d. To get access to d you need $e > d$.

Definition (Malware)**.**

Definition (Mandatory Access Control)**.** *Also known as multilevel security.* MAC is enforced as follows: for each object o, define a set of actions that are applicable to this object, and categorize them by access level. E.g. one could define deletions as *administrator-level* and modifications as *user-level*. The categorization of actions to objects is known as defining the **policy**. The policy is often stored as an **??**. Typical values for o include files, directories, ports, etc. When subject s requests an action a on o, the operating system looks up the rules for o, finds a, and evaluates s against the policy for a on o.

To clarify: in MAC, there is no such thing as "super-user-access", i.e. being a user with high privileges won't necessarily grant you privileges on all objects, because some objects might have policies that only permit certain other users to manipulate them. For example, even though you might be a system administrator, you should not be able to read other user's passwords, nor modify them. You might rather grant sysadmins the privilege to reset the password, and let authenticated users modify their own passwords.

MAC ensures a security policy indepent of user actions. That is, even though a user owns a file, he/she might not be able to do whatever he/she wants with it.

A good example of the usefulness of MAC is that a computer that is hacked may not jepordize other systems. Despite of having obtained some level of control, the attacker may not get access to other critical features.

MLS's are usually difficult to implement. One example is the **cascading** problem; two systems A and B may both have access to an object o. Simeltaneously, system A have access to another object a, system B has access to object b, where $a \neq b$. Now, modifying o might affect how a, b are treated. So if system B wants to alter a, it may modify o, then system A reads o, then modifies a.

Do also note figure 1.6 as it highlights a security caveat of MLS; low-level data can also be worthwhile protecting. See also the quote in Difference between DAC and MAC

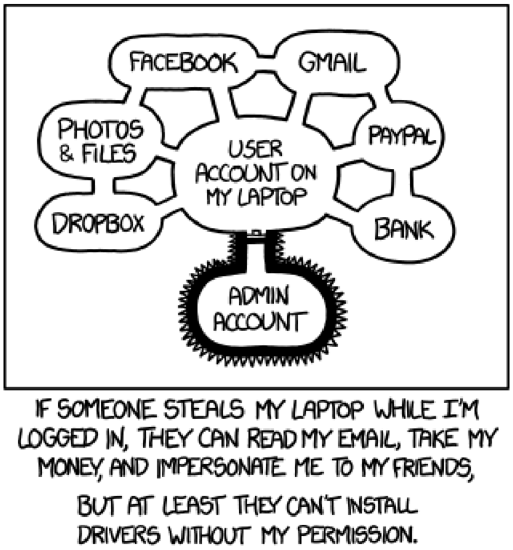

FACEBOOK
GMAIL
PHOTOS & FILES
USER ACCOUNT ON MY LAPTOP
PAYPAL
DROPBOX
BANK
ADMIN ACCOUNT
IF SOMEONE STEALS MY LAPTOP WHILE I'M LOGGED IN, THEY CAN READ MY EMAIL, TAKE MY MONEY, AND IMPERSONATE ME TO MY FRIENDS,
BUT AT LEAST THEY CAN'T INSTALL DRIVERS WITHOUT MY PERMISSION.

Figure 1.2: xkcd.com/1200

Definition (No-operation). NO-OP: From assembly, do nothing.

Definition (Phishing). Essentially the same as Pretexting, just that you're trying to decieve customers instead of staff.

Definition (Pretext). To create a plausible scenario that can decieve personell with sensitive information to disclose this to Eve.

Definition (Principle of least privilege). Programs should only have as much privilege as they need. There's no need to "sudo webbrowser"

Definition (Protection profile). A document that identifies the desired security properties of a product. This is structured as a list of security requirements defined in a way that adheres to Common Criteria

Definition (Pump). A one way transfer (data diode) that takes data from a low access level to a higher access level.

Definition (Race condition)**.** Process A reads resource X, process B reads resource X, process A writes, then process B writes. The value is bad, since B's original value was bad.

Definition (Reflection attacks)**.** First, see Challenge-Response. Suppose that the challenge-response is similar for two entities. I.e. party A sends a challenge N to someone. To solve the challenge, party E simply returns N to A, has party A solve it, and voila, party E knows the solution to N.

Definition (Replay attack)**.** Assume that to login, Alice encrypts her password with one million bits , three times, and sends her password to a login page. The login decrypts and accepts Alice's authentication token.

Eve could simply sniff the POST from Alice and the replay it to the server. Even though the password is highly encrypted, Eve's POST will be valid. The login page accepts Eve.

To mitigate, one should/could implement session tokens; see Session token. As mentioned in that section, however, session tokens also introduce other risks.

Definition (Repudiation attack)**.** To give someone a bad reputation. This could be by rating items with low scores, writing mean reviews, etc.

Definition (Role based access control)**.** Permissions are not related to users, but rather their functions. This means that if a sysadmin is sick, a second-in-rank user could assume the rank of the sysadmin to do whatever is necessary.

Definition (Sandbox)**.** A limited environment. It is common to host websites in sandboxes, so that even if the webserver is hacked, a hacker can't do much on the mainframe.

Definition (Security Policy (model))**.**

Definition (Secrecy)**.**

Definition (Security assurance requirements)**.**

Definition (Security functional requirements)**.**

> *A document that expresses clearly and consicely what the protection mechanisms are to acheieve. . . . It will often take the form of statements about which users may access which data*
>
> — *page 240*

As with most computer policies, it's important the statements are succint, i.e. briefly and clearly expressed, without implications. Also, it's important to make sure that you explicitly define:

- Who determines the policy?
- What qualifies for "need to know"?

- How will the policy be enforced?

Definition (Security requrement). A statement which defines what level of security is utilized for different kind of attacks. It's important that the requirements do not discuss design, only what is required (hence, requirement).

All requirements should be testable. Good ways to do this is to quantify. Quantifying security is difficult, however.

Definition (Security target). A document that describes what a product does, or at the very what it does that has an impact/relevance in security contexts.

Definition (Session token). *Also known as session ID or session identifier.* A session token is a unique identifier, usually in the form of a hash generated by a hash function that is generated and sent from a server to a client to identify the current interaction session.

If Eve can obtain the session ID, she can also, in theory, perform actions, pretending to be the victim. This is known as **session hijacking**. As the session ID needs to be submitted for every POST and GET, it can be easy to obtain it by sniffing or tricking the victim. Hence, there should also be other security measures in place to make the use of session tokens safe.

Definition (Smashing the stack). Making an overflow such that excess bytes are considered as code rather than arguments.

Definition (Software Security Touchpoints).

Definition (SQL-injection). If you can't define this, please go get some sleep.

Definition (Target of Evaluation). ToE. The product under evaluation.

Definition (Trojan). A piece of software that looks cool, but in reality it causes harm when executed.

Definition (Two-channel authentication). I fail to see the big differnece from Two-factor authentication but accoding to the book, this is "sending an access code the user via a separate channel"

Definition (Two-factor authentication). To authenticate, you need "something you have, and something you remember". E.g. a password and a password calculator.

Most companies are sceptical of this. While it does seem to improve security per today's date, it is still prone to real-time mitm attacks.

Definition (validation of input). Given input ι, filter any bad input b and return ι_{clean} There are multiple types of input validation:

> **Blacklist-validation** do not regard context and trim away any bad characters from input. I.e. one can define a set of bad characters in a set $S = \{',", \ \}$, etc. Given input ι, return ι_{clean}, s.t. $\iota_{clean} \cap S = \{\emptyset\}$ The problem with black-validation is that is often easily bypassed, since the attacker can often distort his input in a way that bypasses the filters. Note that the filter only checks for characters, not context.

Whitelist-validation In many contexts, the character ' is considered malicious, however, it is also required in some names, etc. White-validation looks at what structure input should have, and validates accordingly. It is stronger than blacklist-validation in that, for example for dates it is possible to know what structure the input should have , and thereby you can trim from character length and structure.

It is also worhtwile to talk of this as contextual encoding.

Another form of validation is the translation of characters to another typeset. This is typically known as **escaping** input. Before using any input one can e.g. do html-escaping. I assume the reader knows what this is. Character escaping has a recommended order: first do HTML-escaping, then JS-escaping. Finally, it is worth mentioning about escaping that it does not prevent XSS, it just makes it harder to render data as code$^{[why?]}$.

Definition (Valet attack)**.** Assume that you use a random number in your key to unlock something, like a car. This means that whenever you want to unlock the car, there are different codes, all valid.

If Eve gets to unlock Alice's car every day, Eve can record all keys used to unlock Alice's car. Eventually, when Alice's car run out of memory, it will not remember that the first key has ever been used. Eve then tries to use this first key that she recorded. This unlocks the car.

Definition (Virus)**.**

Definition. XSS upload a script that to a server that will be executed by other users. XSS is a popular type of attack and hence there's a lot of terminology. . .

Stored XSS the malicious script sent by an attacker is stored permanently on the webserver.

Reflected XSS the malicious script sent by an attacker causes an immediate malformed response from the server, but the script will not be stored on the server.

DOM-based XSS typically sends a URL to a user such that his/her site, i.e. DOM, is modified. E.g. if the values for a selection field is specified from the url, modifying the parameters to be scripts instead means that the DOM is modified via XSS.

> *Reflected and Stored XSS are server side execution issues while DOM based XSS is a client (browser) side execution issue*
>
> — OWASP, *https://www.owasp.org/index.php/DOM_based_XSS_Prevention_Cheat_Sheet*

Mitigations: see validation of input

1.7 Statistics

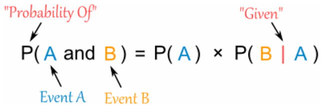
Definition (Bernoulli distribution)**.** A distribution of parameters, where $\forall x, 0 \leq x \leq 1$, Typically dual, e.g. in coin toss there is heads and tails. If $\Pr_{heads} = k$, then $\Pr_{tails} = 1 - \Pr_{heads}$

Definition (Bigvee)**.** $\bigvee\limits_{a \in I} P_a$ says that at least one P_a is true. It can also be used for the maximum value in a set.

Definition (Chernoff bound)**.** In a nutshell, it determines a bound on how many times we must perform a trial to know that our random variables represent a majority. E.g. if we trying to determine that a coin is biased (heads/tails), a chernoff bound will say how many times we must flip the coin to know that we have unraveled a bias. In this case, for example, simply flipping it twice will not suffice.

Definition (Conditional expectations)**.**

$$E(X|Y = y) = \sum_{x \in X} x \ P(X = x|Y = y) = \sum_{x \in X} x \frac{P(X = x, Y = y)}{P(Y = y)}$$

Definition (Conditional probability)**.** 

Definition (Expected value)**.** $E[X] = \sum\limits_{s \in S}^{\dots} X(s) \cdot \Pr(\{s\})$

$E[X] = \sum\limits_{s \in S}^{\dots} X(s) \cdot \Pr(X = x)$

Definition (hyperplane)**.** A plane (surface) that has one less dimension than it's ambient space, i.e. the space around it. E.g. a hyperplane for 3-d dims is only defined 2D.

A hyperplane will therefore act as a separator. Imagine a holding a square in the middle of a ball.

Definition (Indicator variable)**.** Indicator variable: 0 or 1 for whether an element is selected or not.

Definition (Linearity of expectation)**.** $E[X] + E[Y] = E[X + Y]$
$[\sum\limits_{x \in S}^{\dots} X(s) \cdot \Pr(X = x) + \sum\limits_{y \in S}^{\dots} X(s) \cdot \Pr(Y = y)]$
$= [\sum\limits_{s \in S}^{\dots} a \cdot \Pr(Y = a) + a \cdot \Pr(X = a)]$

Theorem 1 (Likelyhood that both X and Y occur in S)**.**
$E[X] \cdot E[Y] = E[X \cdot Y]$

Proof. From Expected value:
$E[X \cdot Y] =$
$\sum_{z \in S}^{\dots} z \cdot \Pr(X = z \text{ and } Y = z) =$
$\sum_{x \in S}^{\dots} \sum_{y \in S}^{\dots} x \cdot y \cdot \Pr(X = x \text{ and } Y = y) =$
$[\sum_{x \in S}^{\dots} x \cdot \Pr(X = x)] \cdot [\sum_{y \in S}^{\dots} y \cdot \Pr(Y = y)] =$
$E[X] \cdot E[Y]$ □

Chapter 2

Algorithms and Methods

2.1 Algorithms

2.1.1 Cristofeledes algorithm

Objective: solve TSP with $\frac{3}{2}$-approximation

1. Construct an MST
2. Extract the odd degree vertices
3. Find a matching, using odd degree vertex from (2)
4. Combine the edges from the tree with those in the matching
5. Form a eularian circuit
6. Short the the traversal path to form a hamiltonian cycle

2.1.2 Double-tree algorithm

2-approximation

1. Construct MST
2. Give each node a redundant entry (graph can now be traversed like an Eularian graph)
3. Traverse the nodes from start to finish(DFS), but only retain the first time occurence of a city

2.1.3 Dijkstra

Assign all nodes distance $= \infty$ Start at given node s. Assign all neighboring nodes their distance from s to n. Proceed to the lowest cost. Repeat. Whenever a better match is found, use the new path instead. The result is a path from s to t, or an MST.

2.1.4 K-center

2-approximation. A form of clustering.

1. Pick arbitrary $i \in V$
2. $S \rightarrow \{i\}$
3. pick other node furthest away from k, add to S
4. repeat, furthest away from all previous picked nodes.

2.1.5 Kruskal's algorithm

Always choose cheapest edge that does not include two pre-discovered nodes. Returns MST.

2.1.6 Nearest addition

2-approximation

1. Find two closest cities
2. Traverse from i to j and back
3. Repeat, consider from each node

2.1.7 Prim's algorithm

Start at a given node s. Choose the cheapest edge. Now repeat, just that you consider s and the node you added. Etc. Never choose an edge that leads to a pre-discovered vertex.

2.1.8 List scheduling algorithm

Whenever a machine is idle, assign it a job. This algorithm runs $\leq 2 \times OPT$.

Proof. Note that $OPT = \frac{\sum p_i}{m}$ We know that the last job, l, starts at time t. Since we've always assigned jobs whenever we could, that means that all machines have so far been busy. Hence we know:

$$t \leq \frac{\sum p_j - p_l}{m} \leq OPT - \frac{p_l}{m}$$

Since OPT is lower bounded as the average work done by each machine.

We can now bound the maximal finishing time:

$$C_{max} \leq t + p_l \leq OPT + p_l \times (1 - \frac{1}{m}) \leq (2 - \frac{1}{m}) \times OPT$$

□

2.1.9 Longest processing time rule

As List scheduling algorithm, just that you first sort the jobs in order of length; put the longest jobs first. This algorithm has approximation ratio $\frac{4}{3} \times OPT$

2.1.10 Shortest remaining time, SRFT

In this scheduling algorithm, the process with the smallest amount of time remaining until completion is selected to execute. This algorithm is applied to Pre-empty schedule schedules

2.1.11 Knapsack DP

Fill out two arrays of $n \times B$, where B is the capacity of the knapsack. In row i, consider item b_i against the capacity given in column j. If $j \geq w_i$, compare the value of $A_{i,j}$ to $A_{i-1,j}$ and use whichever is greater. For all cases where $b_i \geq j$, consider which is better: to use $b_i + A_{i-1,j-w_i}$, or retain the value above. The last equation is: to use current item + whatever we can fit in on the remaining weight, given by the line above. Make sure to simeoltaneously maintain a "keep-array" that gives 1 or zero, indicating whether or not you've used item b_i.

In backtracking, start in the lower right of the keep array. Whenever you get a 1, include item i and decrement i and j by one. Whenever you get a zero, decrement i alone.

The above mentioned runs in $O(n \times W)$, where W is the capasity of the knapsack. The algorithm is *exact*, but pseudopolynomial, since $W = \sum w_i$, which in binary becomes $\log_2 W$, which hence becomes $O(\log_2 W^n)$. An FPTAS, fully polynomial approximation scheme exists, see book.

2.2 Programming Security

2.2.1 Challenge-Response

You insert a car key into a car engine. The engine would now like to know that this key is valid.

Let K be the key, and E the engine. Then,:

$$E \rightarrow K : N \tag{2.1}$$
$$K \rightarrow E : \{E, N\}_K \tag{2.2}$$

Here, N is a challenge sent by the engine to the key in step (1). The key responds by encrypting the text. Iff the engine can decrypt and get valid number, will it start.

2.2.2 Needham-Schroeder Protocol

A primitive, three-way Key-distribution process. Also known to inspire Kerberos[1], the security protocol in windows. The word *primitive* is used to emphasize that the protocol comes from an age where attacks were mostly thought of as external-to-internal penetrations, i.e. one needs to assume that attacks come from external networks, by people without means of authentication. Hence, the protocol fails to defend against users of it's own system.

Having defined key-distribution in section ??, the following should be somewhat intuitive:

$$A \rightarrow S : A, B, N_A \tag{2.1}$$
$$S \rightarrow A : \{N_A, K_{AB}, B, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}} \tag{2.2}$$
$$A \rightarrow B : \{K_{AB}, A\}_{K_{BS}} \tag{2.3}$$
$$B \rightarrow A : \{N_B\}_{K_{AB}} \tag{2.4}$$
$$A \rightarrow B : \{N_B - 1\}_{K_{AB}} \tag{2.5}$$

The difference from plain Key-distribution is that we here introduce the concept of Cryptographic nonces. In step 1, Alices attaches her nonce N_A when talking to Sam. The nonce will be used in Sam's reply. That way, Alice can know that she is not recieving an old/corrupted message.

The entitiy $\kappa = \{K_{AB}, A\}_{K_{BS}}$ represents the the shared key between Alice and Bob, encrypted such that only Bob can read it (denoted by the subscript $_{K_{BS}}$). K_{AB} is the key, and the A is included such that Bob knows it is inteded for his use to Alice. In step 3, Alice sends κ to Bob. Bob can decrypt with his key, given to him earlier by Sam. Finally, to ensure that no data has been corrupted by Eve or other errors, the last two steps verify the legimiticay of the keys. Should Alice fail to return back $N_B - 1$, that is, a nonce from Bob - 1, then Bob can know that he is not talking to a valid user.

[1] originally from Greek/Roman mythology, a three-headed dog, or "hellhound", which guards the entrance of Hades.

One easy to understand flaw in this system is that if someone is able to obtain Alice's identifier, they could easily impersonate Alice. The only required step is to contact Sam, get keys, and then communicate. Alice cannot be aware of this since she never initiated the requests. To revoke the damage done, Sam would have to keep logs of every issued key and thereafter alert everyone who has recieved Alice's key that she was compromised.

Chapter 3

Notes and Thoughts

3.1 Algorithms

The performance guarantee of a primal-dual algorithm provides an upper bound on the Integrality gap of an integer programmin formulation. The Integrality gap also gives a lower bound on the performance guarantee that can be achieved via a standard primal-dual and analysys.

Primal dual is good for when there might be exponential problem sizes in the inout.

General method:

1. Determine decisions
2. Define the value of making those deciosion
3. Decide bounds on OPT
4. Prove that decision bound OPT by α

3.2 Programming Security

It's considered difficult to protect against social engineering. Some of the problem is that you questioning everyone takes time, training everyone takes time, and that since being sceptical takes more time for employees, people arent' going to do it. The best way to protect against it is to never trust anyone, and build autonomous systems.

The problem about centralizing data is that once a breach is made, the effects can be more severe. If a small datasource is compromised occasionally, the negative impacts might still be small enough to be handled. However, it is easier making centralized systems safe against small crimes.

Passwords are good to authenticate, but they're easy to guess and people are not good at ensuring policies for passwords. When you ask for secure passwords, people tend to store them in an unsafe fashion such that the **security at the expense of usability is usability at the expense of security**.

In general, protocols may be defeated byy changing the environment that they operate in. Most protocols make some assumptions about how things work - as soon as this is modified, the protocol might break.

A cool form of attack: if a privileged user has "." in his path, and you could put in a piece of software there, like "ls", you could trick the admin into executing it. Since the "ls" in "." might be preferred, he will then launch a program with elevated privileges, that you might have put there.