Polynomial time

ian.mcloughlin@gmit.ie

Languages recap

Alphabet is a set usually denoted A.
e.g. $A = \{0, 1\}$

Kleene star of a an alphabet is all strings over it.
e.g. $A^* = \{\epsilon, 0, 1, 00, 01, \ldots\}$

Language is a subset of the Kleene star of an alphabet.
e.g. $L = \{10, 11, 101, 111, 1011, \ldots\} \subseteq A^*$

Turing machine accepts a language. It decides if it always halts.

TIME

$TIME(f(n))$ is the set of languages with an $O(n^k)$ decider.

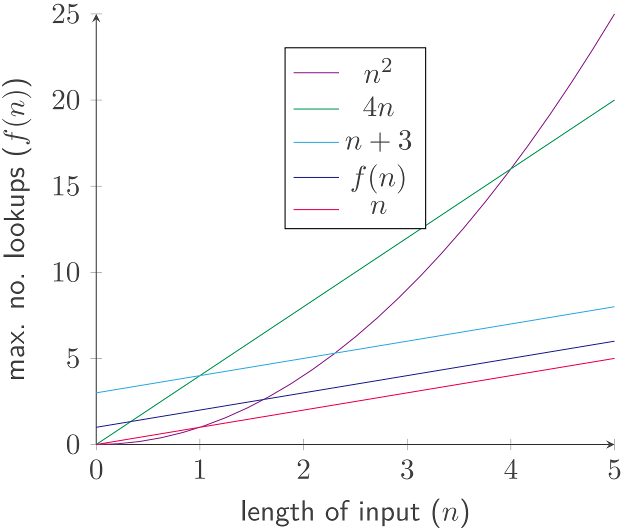
Denote by n the length ($\in \mathbb{N}$) of an input string to a decider (Turing machine that always halts).

Calculate the maximum number of state table lookups a given decider takes to halt in terms of n.

One way to do this is calculate the value for $n = 0$, $n = 1$, $n = 2$, $n = 3$, and so on, and try to generalise.

Find a succinct function $f(n)$ that bounds it beyond a fixed value of n. Then we say the decider is $O(f)$. Note that deciders are $O(g)$ for lots of functions g.

Big-O



25
20
15
10
5
0
0
1
2
3
4
5
n^2
$4n$
$n + 3$
$f(n)$
n
max. no. lookups ($f(n)$)
length of input (n)

Polynomial time

$$P = \bigcup_k \mathsf{TIME}(n^k)$$

Decidable languages are languages for which there is at least one Turing machine that halts in a finite number of state table lookups for each input.

P is the set of languages that are decidable in polynomial time using a deterministic Turing machine.

Polynomial means that for a length of input n the number of steps (state table lookups) is $O(n^k)$ for some $k \in \mathbb{N}$.

TIME(n^k) is the set of languages decidable in $O(n^k)$ steps.

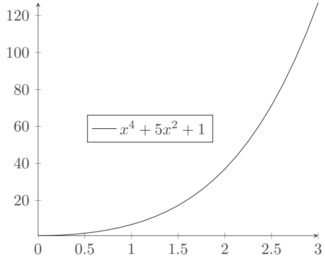
Recap on polynomials

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m \text{ where } a_i \in \mathbb{R}, m \in \mathbb{N}$$

Polynomials have a fixed number of operations (add, multiply) to perform for all values of x.

Exponential contrasts with polynomial, having a variable number of operations, e.g. 2^x.

Functions that are polynomial in n are functions that take n and plug it into a polynomial to give the result.



Polynomials are closed

$$\mathbb{R}[x] = \{a_0 + a_1x + a_2x^2 + \cdots + a_mx^m \mid a_i \in \mathbb{R}, m \in \mathbb{N}\}$$

The set of all polynomials is closed under the following operations:

Addition: $(x^4 + 1) + (5x^3 + x) = x^4 + 5x^3 + x + 1$

Multiplication: $(x^4 + 1) \times (5x^3 + x) = 5x^7 + x^5 + 5x^3 + x$

Composition: $((5x^3 + x))^4 + 1 =$
$625x^{12} + 500x^{10} + 150x^8 + 20x^6 + x^4 + 1$

Applying two polynomial time algorithms one after the other is still polynomial time.

Tractability

Cobham's thesis is that P represents the tractable problems.

Operations $+$, $-$, $\times$ and $\div$ are $O(n^k)$.

Space – a decider for a problem in P cannot move more than a polynomial number of steps along the tape.

Exponential time is a superset of polynomial time.

Subexponential means the language is in a complexity class less than exponential time.

Strictly exponential time problems are considered intractable.

Careful: would you consider a language whose best known decider is $O(n^{100000})$ tractable?

The exponential time hypothesis is that 3-SAT can't be solved in subexponential time. 3-SAT is NP-complete.