Part I

Complexity Measures

1 Deterministic measures

Let T be an algorithm type. For an algorithm A of type T, it must be defined, on input x

- when A terminates
- what the result is

An algorithm A of type T *computes* a mapping $\varphi_A : (\Sigma^*)^m \to \Sigma^*$ with

$$\varphi_A(x) := \begin{cases} \text{result of } A \text{ on } x & \text{if } A \text{ terminates} \\ \text{undefined} & \text{otherwise} \end{cases}$$

Def. A *complexity measure* for A of type T is a mapping

$$\Phi : \text{finite computation of } A \text{ of type } T \text{ on } x \mapsto r \in \mathbb{N}$$

Ex. examples for Φ are T-DTIME, T-DSPACE

Def. A *complexity function* of A of type T is a mapping $\Phi_A : (\Sigma^*)^M \to \mathbb{N}$ with

$$\Phi_A(x) := \begin{cases} \Phi(\text{computation of } A \text{ on } x) & \text{if } A \text{ terminates} \\ \text{undefined} & \text{otherwise} \end{cases}$$

Def. *worst-case complexity function*: $\Phi_A : \mathbb{N} \to \mathbb{N}$ with

$$\Phi_A(n) := \max_{|x|=n} \Phi_A(x)$$

"worst" complexity of all inputs of this length

Def. Algorithm A of type T *computes* f *in* Φ*-complexity* t iff

$$\varphi_A = f \text{ and } \Phi_A \leq_{ac} t$$

Note

- If not specified, an "algorithm" is short for "algorithm A of type T".
- Alphabets are arbitrary but finite
- Numbers are encoded dyadically

Ex. Some complexity classes of *functions*:

- $F\Phi(t) := \{f \mid f \text{ is total function and there ex. alg. } A \text{ of type } T \text{ computing } f \text{ in } \Phi\text{-complexity } t\}$
- $F\Phi(\mathcal{O}(t)) := \bigcup_{k\geq 1} F\Phi(k \cdot t)$
- $F\Phi(\text{Pol } t) := \bigcup_{k\geq 1} F\Phi(t^k)$

Ex. Complexity classes of *languages*:

- $\Phi(t) := \{L \mid L \subseteq \Sigma^*$ and there ex. alg. accepting L in Φ-complex
- $\Phi(\mathcal{O}(t))$ similar to definition for functions
- $\Phi(\text{Pol } t)$ likewise

We now turn to complexity measures for Turing Machines.

- TMs are characterised by one read-only input tape, an arbitrary number of working tapes and a write-only output tape. The input tape can be *one-way* infite or *two-way* infinite.
- The *input* is the full initial configuration of input tape and all other tapes.
- If a TM M stops, the output is given by the leftmost consecutive word on the output tape.

Def Algorithm types for turing machines:

T	one w.t.	k w.t	arbitrarily many w.t.
no input tape	T	kT	multi T
one-way inp. t.	$1-T$	$1-kT$	$1-$multi T
two-way inp. t.	$2-T$	$2-kT$	$2-$multi T

Def Let M be a T-TM, let β be a computation of $M(x)$.

- T-DTIME(β) := number of steps of β
- T-DTIME(β) := number of cells of write tape visited (or non-empty)

We can define complexity classes based on the characteristics of the employed TM:

$$(F)\begin{Bmatrix} 0 \\ 1 \\ 2 \end{Bmatrix} - \begin{Bmatrix} T \\ kT \\ \text{multi}T \end{Bmatrix} - \begin{Bmatrix} \text{DTIME} \\ \text{DSPACE} \end{Bmatrix}(r)$$

Ex.

- len : $x \mapsto |x| \in F1{-}T\text{-DTIME}((1+\varepsilon)n)$ for all $\varepsilon > 0$, where $|x| = n$