1. Complexity measures and classes

1.1 Deterministic measures

1.1.1 General framework

- let τ be an algorithm type (e.g., TM, RAM, Pascal/C/Java, ...)
- for alg. A of type τ, it must be defined on input x
  - when A terminates (stops, halts)
  - what the result (outcome, output) is
- alg. A of type τ computes a mapping $\varphi_A : (\Sigma^*)^m \to \Sigma^*$:

$$\varphi_A(x) =_{def} \begin{cases} \text{result of } A \text{ on } x & \text{if } A \text{ terminates on } x \\ \text{n. d.} & \text{otherwise} \end{cases}$$

- complexity measure for alg A of type τ is a mapping

Φ : finite comp. of A of type τ on x $\mapsto$ $r \in \mathbb{N}$

Examples: standard complexity measures:

$\Phi = \tau - DTIME$, $\Phi = \tau - DSPACE$

„D" indicates determinism

- complexity function of A of type τ is a mapping $\Phi_A : (\Sigma^*)^m \to \mathbb{N}$:

$$\Phi_A(x) =_{def} \begin{cases} \Phi(\text{computation of } A \text{ on } x) & \text{if } A \text{ terminates on } x \\ \text{n. d.} & \text{otherwise} \end{cases}$$

- worst-case complexity function of A of type τ: $\Phi_A : \mathbb{N} \to \mathbb{N}$:

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

- let $t: \mathbb{N} \to \mathbb{N}$ be a resource bound (i.e., t is monotone)
- alg. A of type τ computes a total function f with $\underline{\Phi}$-complexity t iff $\varphi_A = f$ and $\underline{\Phi} \leq_{ae} t$.
  ($g \leq_{ae} h \Leftrightarrow_{def} (\exists n_0)(\forall n \geq n_0)[\, g(n) \leq h(n) \,]$)
- define following complexity classes of functions:

$$F\underline{\Phi}(t) =_{def} \{ f \mid f \text{ is a total function and there is an alg. } A \text{ of type } \tau \text{ computing } f \text{ in } \underline{\Phi}\text{-complexity } t \}$$

$$F\underline{\Phi}(O(t)) =_{def} \bigcup_{k \geq 1} F\underline{\Phi}(k \cdot t)$$

$$F\underline{\Phi}(\mathrm{Pol}\ t) =_{def} \bigcup_{k \geq 1} F\underline{\Phi}(t^k)$$

- define following complexity classes of languages:

$$\underline{\Phi}(t) =_{def} \{ L \mid L \subseteq \Sigma^* \text{ and there is an alg. } A \text{ of type } \tau \underbrace{\text{accepting}}_{\varphi_A = c_L} L \text{ in } \underline{\Phi}\text{-complexity } t \}$$

$$\underline{\Phi}(O(t)) =_{def} \bigcup_{k \geq 1} \underline{\Phi}(k t)$$

$$\underline{\Phi}(\mathrm{Pol}\ t) =_{def} \bigcup_{k \geq 1} \underline{\Phi}(t^k)$$

Remarks:

(1.) alphabets may be arbitrary but finite

(2.) numbers are encoded in dyadic:

- for $L \subseteq \mathbb{N}^m$: $L \in \underline{\Phi}(t) \Leftrightarrow_{def} \{ (dya(n_1), \dots, dya(n_m)) \mid (n_1, \dots, n_m) \in L \} \in \underline{\Phi}(t)$
- $f: \mathbb{N}^m \to \mathbb{N}$:

$$f \in F\underline{\Phi}(t) \Leftrightarrow_{def} f' \in F\underline{\Phi}(t) \text{ where } f': (\{1,2\}^*)^m \to \{1,2\}^*$$

$$f'(x_1, \dots, x_m) =_{def} dya(f(dya^{-1}(x_1), \dots, dya^{-1}(x_m)))$$

- dya: $\mathbb{N} \to \{1,2\}^*$ recursively defined by

$$dya(0) =_{df} \varepsilon$$

$$dya(2n+1) =_{df} dya(n)1$$

$$dya(2n+2) =_{df} dya(n)2$$

- $dya^{-1} : \{1,2\}^* \to \mathbb{N}$ given by

$$dya^{-1}(a_{n-1} \dots a_1 a_0) = \sum_{k=0}^{n-1} a_k 2^k$$

- dya is bijective (one-to-one and onto)

1.1.2 Complexity measures for RAMs

- τ = RAM (random access machine)
- registers R0, R1, R2, ..., each reg. Ri containing number $\langle R_i \rangle \in \mathbb{N}$
- instruction register BR containing next instr. $\langle BR \rangle$
- instruction set: $Ri \leftarrow Rj$, $RRi \leftarrow Rj$, $Ri \leftarrow RRj$ (transport)

  $Ri \leftarrow k$, $Ri \leftarrow Rj \pm Rk$ (arithmetic)

  GOTO k, IF Ri=0 GOTO k (jumps)

  STOP
- input $(x_1, \dots, x_m) \in \mathbb{N}^m$ given by foll. initial configuration:

$$\langle Ri \rangle := x_{i+1} \quad \text{for } 0 \le i \le m-1$$

$$\langle Ri \rangle := 0 \quad \text{for } i \ge m$$

- comp. stops if $\langle BR \rangle = 0$, output is given by $\langle R0 \rangle$
- let β be a comp. of a RAM (i.e., program of a RAM):

$$\text{RAM-DTIME}(\beta) =_{df} \text{number of steps (tacts, cycles) of } \beta$$

$$\text{RAM-DSPACE}(\beta) =_{df} \max_{t \ge 0} BIT(\beta, t)$$

where

$$BIT(\beta, t) =_{df} \sum_{i \ge 0} |dya(\langle Ri \rangle_t)| + \sum_{\langle Ri \rangle_t \neq 0} |dya(i)|$$

and $\langle Ri \rangle_t$ is number stored in Ri after step t of β

Example: RAM M for comp. $mult : \mathbb{N} \times \mathbb{N} \to \mathbb{N} : (x,y) \mapsto x \cdot y$

[1] $R3 \leftarrow 1$

[2] IF $R1 = 0$ GOTO 6

[3] $R2 \leftarrow R2 + R0$

[4] $R1 \leftarrow R1 - R3$

[5] GOTO 2

[6] $R0 \leftarrow R2$

[7] STOP

time complexity: on input (x,y), M takes $4y + 4$ steps,
$n = |dya(x)| + |dya(y)|$

using $2^{|dya(x)|} - 1 \leq x \leq 2^{|dya(x)|+1} - 2$, we obtain: $(x=0)$

$$
\begin{aligned}
4y + 4 &\leq 4 \cdot (2^{|dya(y)|+1} - 2) + 4 \\
&\leq 4 \cdot (2^{n+1} - 2) + 4 \qquad (n = |dya(x)| + |dya(y)|) \\
&= 2^{n+3} - 4
\end{aligned}
$$

So, $\text{RAM-DTIME}_M(n) = 2^{n+3} - 4$ and

$mult \in \text{FRAM-DTIME}(O(2^n))$

1.1.3 Complexity measures for Turing machines

read-only input tape (one-way, two-way)

$k \in \mathbb{N}$ working tapes (no restrictions)

write-only output tape

algorithm types:

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

- input $(x_1, \dots, x_n)$ is given by foll. initial configurations:

$$\dots \square\square\, x_1 * x_2 * \dots * x_n \,\square\square\square \dots \qquad \text{i.t.}$$

$$\dots \square\square\square \dots \qquad \text{all other tapes}$$

- if M stops, output is given by the configuration

$$\dots \square\square\square\, z \,\square\square\square \dots \qquad \text{output}$$

where z is the leftmost word not containing $\square$

- let M be a τ-TM, let β be a computation of $M(x)$:

τ-DTIME(β) $=_{df}$ number of steps of β

τ-DSPACE(β) $=_{df}$ number of cells on w.t. visited (or containing any symbol other than $\square$ during β)

$$\tau\text{-DTIME}_M(x) =_{df} \begin{cases} \text{number of steps of comp. of } M(x) & \text{if } M(x) \text{ terminates} \\ \text{n.d.} & \text{otherwise} \end{cases}$$

$$\tau\text{-DSPACE}_M(x) =_{df} \begin{cases} \text{number of cells on w.t. visited or cont. any symbol other than } \square \text{ during } M(x) & \text{if } M(x) \text{ terminates} \\ \text{n.d.} & \text{otherwise} \end{cases}$$

- Complexity class w.r.t. resource bound τ:

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

Examples:

(1) For function len: $x \mapsto |x|$, len $\in$ F1-T-DTIME$((1+\varepsilon)n)$ for all $\varepsilon > 0$.

(2) Consider foll. context-free languages over $\Sigma = \{0,1\}$:

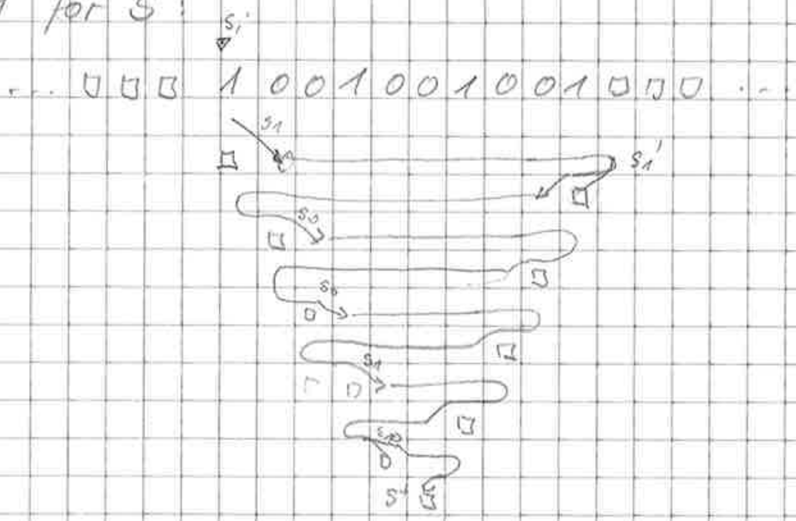
$$S =_{df} \{ x x^R \mid x \in \{0,1\}^* \}$$

$$C =_{df} \{ 0^n 1^n \mid n \in \mathbb{N} \}$$

Complexity classes of S, C specified by r:

	S	C
T-DTIME (r)	$\varepsilon \cdot n^2$	$\varepsilon \cdot n \log n$
1-T-DTIME (r)	$\varepsilon \cdot n^2$	n
2-T-DTIME (r)	$(1{,}5+\varepsilon)n$	n
1-T-DSPACE (r)	$\varepsilon \cdot n$	$\varepsilon \cdot \log n$
2-T-DSPACE (r)	$\varepsilon \log n$	$\varepsilon \cdot \log n$

- T-TM for S:



- analysis for $|x| = 2m$:

$$\text{T-TIME}_M(x) \le \sum_{i=0}^{m-1} \left(2(m-i) + 2(m-i) + 1\right)$$

$$= \sum_{i=0}^{m-1} \left(4(m-i) + 1\right)$$

$$= \sum_{i=1}^{m} (4i+1) = 4\frac{m(m+1)}{2} + m$$

$$= 2m^2 + 3m \le_{ae} \frac{2}{3}|x|^2$$

- now instead of comparing on symbol, compare k symbols (expanding state set):

$$\text{T-TIME}_M(x) \le \left(\sum_{i=0}^{\lfloor m/k \rfloor} (|x| - 2ki) + (|x| - 2ki - (k-1)) + 1\right) + k$$

$$= \left(\sum_{i=0}^{\lfloor m/k \rfloor} 4m - (4i+1)k + 2\right) + k$$

$$\le (4m+1)\left(\frac{m}{k} + 1\right) + k \le_{ae} \frac{2}{k}|x|^2$$

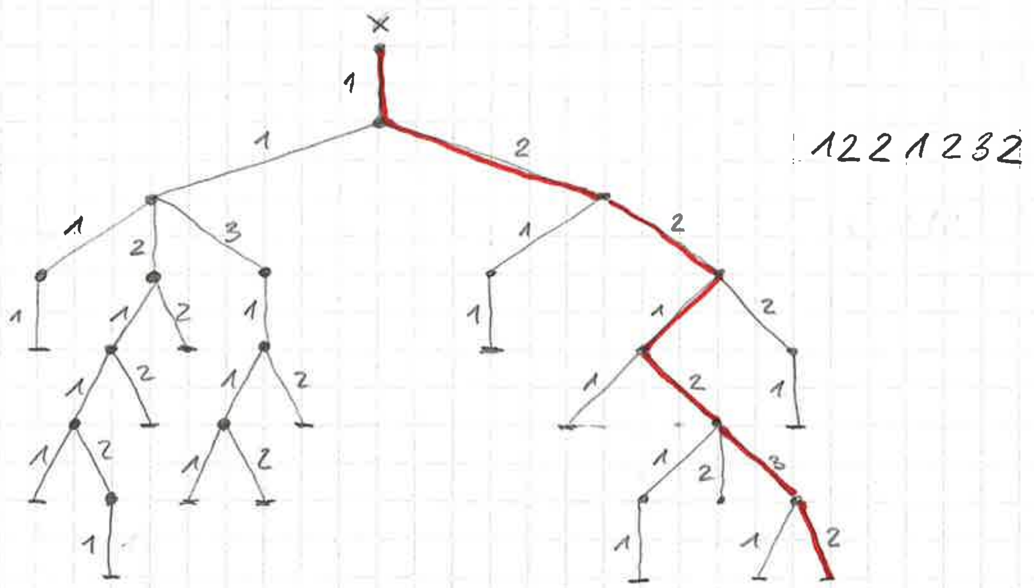
1.1.2 Nondeterministic measures

nondeterminism of alg. is a comp. mode:

- possibly many instructions to perform next
- realized in parallel (by identical copies of mach./alg.)
- number of parallel instructions limited by a constant number

nondeterministic RAM: instructions may have same numbers
nondeterministic TM: many transitions applicable given a conf.

Computation tree:



Let k be max. number of nondet. branches of M.
A computation path of M on x is a word $a_1 \dots a_T \in \{1, \dots, k\}^*$ s.t. a_t is the a_t-th instruction of the same number in t-th step of comp.
Note that not each word from $\{1, \dots, k\}^*$ describes a comp. path.

Define „acceptance by a nondet. machine":

- τ is alg. type
- A is a nondet. alg. of type τ
- x is an input
- z is a comp. path of A on x

We define

- $\varphi_A(x|z) =_{def}$ result of A on x along z
- A accepts x $\Leftrightarrow_{def}$ there is a z of A on x s.t. $\varphi_A(x|z)=1$
  (i.e., there is an accepting comp. path of A on x)
- A accepts $L \subseteq \Sigma^*$ $\Leftrightarrow_{def}$ $L = \{x \in \Sigma^* \mid A \text{ accepts } x\}$

Complexity measure for nondet. alg. A of type τ:

Φ: comp. path of A of z on x $\mapsto r \in \mathbb{N}$

Example: $\Phi = \tau\text{-NTIME}$ (nondeterm. time)
$\Phi = \tau\text{-NSPACE}$ (nondeterm. space)

Note: if A is actually deterministic then

$\tau\text{-NTIME}$ (comp. path) $=$ $\tau\text{-DTIME}$ (computation)
$\tau\text{-NSPACE}$ (comp. path) $=$ $\tau\text{-DSPACE}$ (computation)

Complexity functions for nondet. alg.

$$\Phi_A(x|z) =_{def} \Phi(\text{comp. path } z \text{ of } A \text{ on } x)$$

$$\Phi_A(x) =_{def} \min\{\Phi_A(x|z) \mid \varphi_A(x|z)=1\} \quad (\min \emptyset =_{def} 0 \,!)$$

$$\Phi_A(n) =_{def} \max\{\Phi_A(x) \mid |x|=n\}$$

Nondeterministic complexity classes:

- A accepts L in Φ-complexity t $\Leftrightarrow_{def}$ A accepts $L \wedge \Phi_A \leq_{ae} t$
- $\Phi(t) =_{def} \{ L \mid L \subseteq \Sigma^* \text{ and there is a nondet. alg. } A$ of type τ accepting L in Φ-complexity $t \}$
- $\Phi(O(t)) =_{def} \bigcup_{k \geq 1} \Phi(k \cdot t)$
- $\Phi(\text{Pol } t) =_{def} \bigcup_{k \geq 1} \Phi(t^k)$

Proposition 2.

Let τ be any alg. type, t, t' be resource bounds.

(1.) $t \leq_{ae} t' \Rightarrow \Phi(t) \subseteq \Phi(t')$ for all Φ

(2.) $\tau\text{-DTIME}(t) \subseteq \tau\text{-NTIME}(t)$

(3.) $\tau\text{-DSPACE}(t) \subseteq \tau\text{-NSPACE}(t)$

Remark: For det. complexity classes, the followin equiv. are true:

$L \in \tau\text{-DTIME}(t) \Leftrightarrow \overline{L} \in \tau\text{-DTIME}(t)$

$L \in \tau\text{-DSPACE}(t) \Leftrightarrow \overline{L} \in \tau\text{-DSPACE}(t)$

For τ-NTIME and τ-NSPACE, this need not be true.

Example: $C =_{def} \{ 0^n 1^n \mid n \in \mathbb{N} \}$; we show $\overline{C} \in T\text{-NTIME}(n \cdot \log n)$

Define M to be the T-TM that, on input x,

(a) guesses a $k \in \mathbb{N}_+$

(b) determines $\alpha_k =_{def} \mathrm{mod}(|x|_0, k)$

(c) determines $\beta_k =_{def} \mathrm{mod}(|x|_1, k)$

(d) accepts if and only if $\alpha_k \neq \beta_k$ or there occurs a 1 foll. by 0

Clearly, M accepts $\overline{C}$.

Complexity analysis of M on x, $|x| = 2n$:

- $x \notin \overline{C}$ (i.e., $x = 0^n 1^n$): no accepting path.
- $x \in \overline{C}$ (i.e., $|x|_0 \neq |x|_1$): there is an accepting path z_k for $k \leq n$, thus

$$T\text{-}NTIME_M(x/z_k) \leq |x| \cdot \log k$$

So, $\overline{C} \in T\text{-}NTIME(n \log n)$

Finer analysis use Chinese remainder theorem ($\mathbb{Z}_{pq} \cong \mathbb{Z}_p \times \mathbb{Z}_q$; more specifically: for l natural numbers $b_1, \dots, b_l$, prime numbers $p_1, \dots, p_l$ there is exactly one $x \in \mathbb{Z}_{p_1 \cdots p_l}$ s.t. $x \equiv b_i \pmod{p_i}$ for all $i \in \{1, \dots, l\}$);

$x \in \overline{C}$

we find $\alpha_k \neq \beta_k$ for at least one of the first prime numbers $p_1, \dots, p_m$ s.t. $n \leq p_1 \cdot \ldots \cdot p_m$.

So, $m =_{ae} \lceil \log n \rceil$ is enough ($n \leq 2^{\log n} \leq 2^m \leq p_1 \cdot \ldots \cdot p_m$).

By prime number theorem, we know $p_m \leq c \cdot m \cdot \log m$ for $c > 1$. We conclude

$$\begin{aligned} T\text{-}NTIME_M(x/z_k) &\leq |x| \cdot \log k \\ &\leq |x| \cdot \log p_m \\ &\leq |x| \cdot \log(c \cdot \log|x| \cdot \log\log|x|) \\ &\leq 2|x| \cdot \log\log|x| + |x| \cdot \log c \\ &<_{ae} 3|x| \cdot \log\log|x| \end{aligned}$$

Hence, $\overline{C} \in T\text{-}NTIME(n \cdot \log\log n)$

1.1.3 Type-independent measures

Theorem 3.

For $X \in \{D, N\}$ and all $s(n) \geq 0$, the following holds:

(1.) $i\text{-}kT\text{-}XSPACE(s) = i\text{-}multiT\text{-}XSPACE(s)$ for $i = 0,1,2$, $k \geq 1$

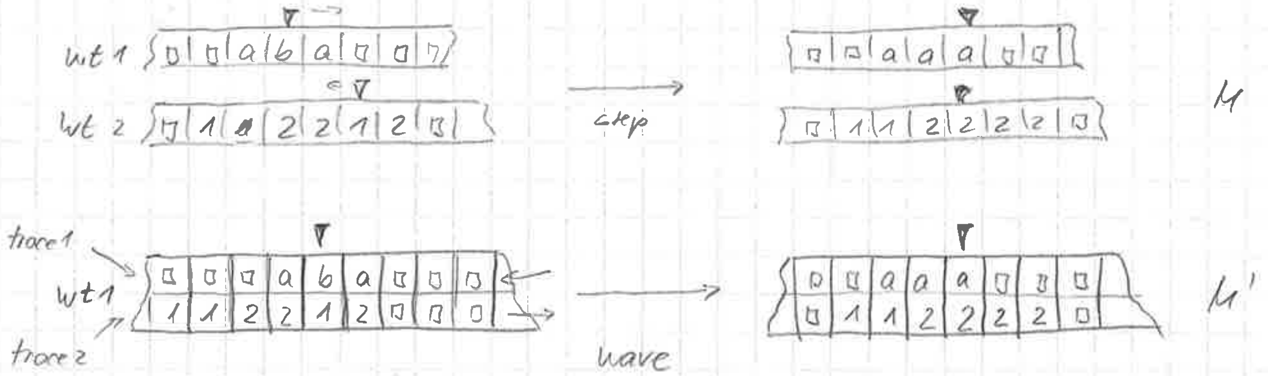
(2.) $T\text{-}XSPACE(s) = 1\text{-}T\text{-}XSPACE(s) = 2\text{-}T\text{-}XSPACE(s)$ for $s(n) \geq n$

(3) $1\text{-}T\text{-}XSPACE(s) \subseteq 2\text{-}T\text{-}XSPACE(s)$

(4.) $RAM\text{-}XSPACE(O(s)) = T\text{-}XSPACE(s)$ for $s(n) \geq n$

Proof: (Sketches)

(1.) Use vector alphabets and traces:



Fix heads and move traces in opposite directions comp. to original machine M.

(2) Inclusion chain:

- $T\text{-}XSPACE(s) \subseteq 1\text{-}T\text{-}XSPACE(s)$: copy input to working tape of a 1-T-TM
- $1\text{-}T\text{-}XSPACE(s) \subseteq 2\text{-}T\text{-}XSPACE(s)$: Trivial
- $2\text{-}T\text{-}XSPACE(s) \subseteq T\text{-}X\text{-}SPACE(s)$: Simulate input tape and working tape by traces of T-TM

(3) Trivial.

(4) Proof omitted. ∎

Space-complexity classes (for $s(n) \geq 0$)

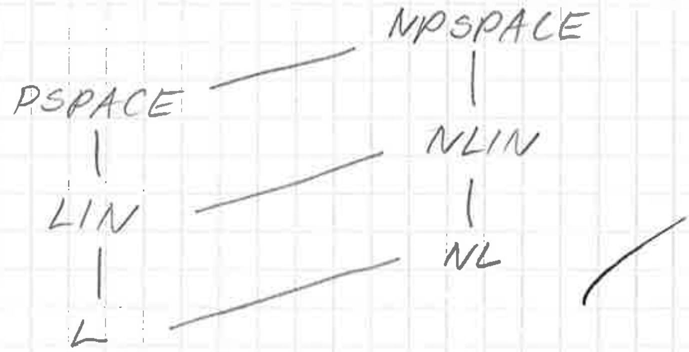
$$DSPACE(s) =_{def} 2\text{-}T\text{-}DSPACE(s)$$
$$NSPACE(s) =_{def} 2\text{-}T\text{-}NSPACE(s)$$

Note: 2-way input tape most flexible in sublinear space

Special space-complexity classes:

$$L =_{def} DSPACE(\log n)$$
$$NL =_{def} NSPACE(\log n)$$
$$LIN =_{def} DSPACE(O(n))$$
$$NLIN =_{def} NSPACE(O(n))$$
$$PSPACE =_{def} DSPACE(Pol\ n)$$
$$NPSPACE =_{def} NSPACE(Pol\ n)$$
$$(EXPSPACE =_{def} DSPACE(2^{Pol\ n}))$$

Inclusions (Hasse diagram):



Remarks:

1. For arbitrary $s: \mathbb{N} \to \mathbb{N}$, it is open whether $DSPACE(s) \subset NSPACE(s)$ or whether $DSPACE(s) = NSPACE(s)$
2. Special open questions: $L \overset{?}{=} NL$, $LIN \overset{?}{=} NLIN$ (aka the first LBA problem)

Theorem 4.

For $X \in \{D, N\}$ and all $t(n) \geq n$, the following holds:

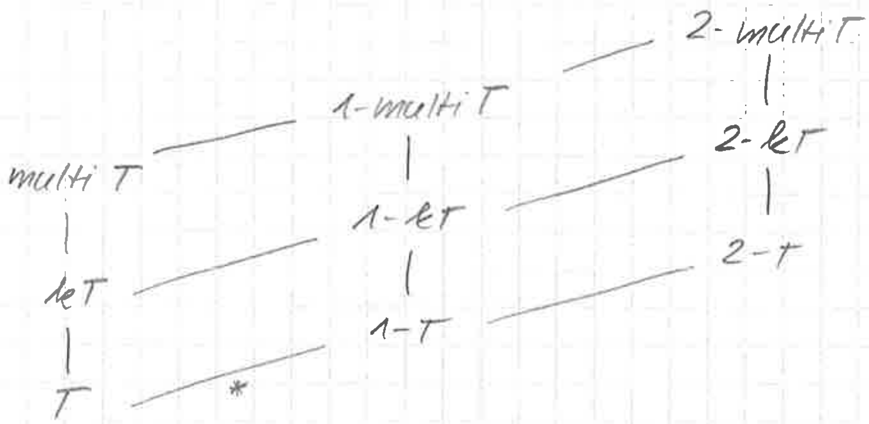
(1.) i- kT - XTIME (Pol t) = i- multiT - XTIME (Pol t) for $i \in \{0,1,2\}$, $k \geq 1$

(2.) T - XTIME (Pol t) = 1-T-XTIME (Pol t) = 2-T-XTIME (Pol t)

(3.) RAM - XTIME (Pol t) = T-XTIME (Pol t)

Proof: (Sketch)

Trivial inclusions for same time bound t:



*: T-XTIME (t) ⊆ 1-T-XTIME (t): Copy new input symbol from input tape to working tape of 1-T-TM if accessed for the first time

(1.), (2.): It suffices to show: 2-multiT-XTIME (t) ⊆ T-XTIME ($O(t^2)$). Use same construction as in the space case: maximum amount of cells $\leq c \cdot t(n)$; each step of old machine M is simulated by $c' \cdot t(n)$ steps of new T-TM M' for readjusting the traces, so time complexity of M' is $\leq_{ae} c'' \cdot t(n)^2$ for some $c'' \geq c \cdot c'$.

traces → tracks

(3.) Proof omitted. ■

time-complexity classes (for $t(n) \geq n$):

$$DTIME(Pol\ t) =_{def} T\text{-}DTIME(Pol\ t)$$
$$NTIME(Pol\ t) =_{def} T\text{-}NTIME(Pol\ t)$$

special time-complexity classes:

$$P =_{def} DTIME(Pol\ n)$$
$$NP =_{def} NTIME(Pol\ n)$$
$$E =_{def} DTIME(Pol\ n)$$
$$NE =_{def} NTIME(Pol\ n)$$
$$EXP =_{def} DTIME(2^{Pol\ n})$$
$$NEXP =_{def} NTIME(2^{Pol\ n})$$

Inclusions:

$$\begin{tikzcd}
 & NEXP \arrow[d, dash] \\
EXP \arrow[ur, dash] \arrow[d, dash] & NE \arrow[d, dash] \\
E \arrow[ur, dash] \arrow[d, dash] & NP \\
P \arrow[ur, dash] &
\end{tikzcd}$$

Remarks:

1. For arbitrary $t: \mathbb{N} \to \mathbb{N}$, it is open whether $DTIME(t) \subset NTIME(t)$ or whether $DTIME(t) = NTIME(t)$; exception: $DTIME(O(n)) \subset NTIME(O(n))$
2. Special open questions: $P \stackrel{?}{=} NP$, $E \stackrel{?}{=} NE$, $EXP \stackrel{?}{=} NEXP$

Remarks:

There are some results regarding tight simulations for „easy-to-comp." time bounds $t(n) \geq n$:

- $\text{multiT-DTIME}(t) \subseteq \text{T-DTIME}(t^2)$
- $\text{multiT-DTIME}(t) \subseteq \text{2T-DTIME}(t \log t)$
- $\text{multiT-NTIME}(t) = \text{2T-NTIME}(t)$
- $\text{multiT-DTIME}(t) \subseteq \text{RAM-DTIME}(O(t/\log t))$ for $t(n) \geq n \log n$
- $\text{RAM-DTIME}(t) \subseteq \text{multiT-DTIME}(t^3)$
- $\text{multiT-NTIME}(t) \subseteq \text{RAM-NTIME}(O(t)) \subseteq \text{multiT-NTIME}(t^3)$

Proof idea for $\text{multiT-NTIME}(t) \subseteq \text{2T-NTIME}(t)$:

kT-NTM M is simulated by 2T-NTM M' in $k+1$ phases:

- phase 0: guess all symbols of all tapes for each time step and all nondeterministic choices of M
- phase $i>0$: on working tape 2 of M', simulate the work of M on tape i by using guessed symbols on tape 1; check whether content on tape 1 is correct, if not reject input and stop

If no rejection after phase k then sequence of symbols was correctly guessed; so, M' accepts input iff M accepts on guessed computation path.

Running time: $(k+1) \cdot c \cdot t(n)$