Our TM-model

An <u>ordinary TM</u> is a TM with <u>one</u> tape, or <u>string</u> as we will say.

- the left end of the string always contains the symbol '$\triangleright$' which is never overwritten and the cursor never moves to the left of $\triangleright$.
- input = finite string beginning with $\triangleright$.
- the cursor may move to the right of the string and in this case we see it as appending a blank symbol '$\sqcup$' to the end of the string. <u>Thus the string never gets shorter.</u>
- There are three halting states: "yes", "no" and h.

A <u>k-string TM</u> $(k \geq 1)$ is one with k strings; on each the TM works a described above.

- input is always placed on string 1.
- output, if any, always appears on string k.

If x is input then eventual output is denoted by $M(x)$.

We say that a TM M <u>decides</u> the language L if

$$x \in L \Rightarrow M(x) = \text{"yes"},$$
$$x \notin L \Rightarrow M(x) = \text{"no"}.$$

(and M <u>accepts</u> L if

$$x \in L \Rightarrow M(x) = \text{"yes"},$$
$$x \notin L \Rightarrow M(x) \text{ does not halt,}$$
denoted $M(x) = \nearrow$.)

Time bounds

Def. $L \in TIME(f(n)) \iff$
L is decided by a k-string TM ($k \geq 1$) within time $f(n)$, i.e. for input x the TM needs at most $f(|x|)$ steps.

Thm 2.1 For every k and k-string TM M operating within time $f(n)$, there is a 1-string TM (an "ordinary TM") M' operating within time $O(f(n)^2)$ such that $M(x) = M'(x)$ for every input x.

Proof idea: Split the string/tape of M' the 1-string TM into k parts, corresponding to the k strings of M.

Constants don't matter for time complexity classes:

<u>Thm 2.2</u> Let $L \in TIME(f(n))$. Then, for every $\varepsilon > 0$, $L \in TIME(\varepsilon f(n) + n + 2)$.

<u>Proof idea</u> Let M decide $L \in TIME(f(n))$. Let $m = \lceil \frac{6}{\varepsilon} \rceil$. Construct M' simulating M but with an extended alphabet containing (in addition) a symbol encoding every string of length m from M:s alphabet.

Space bounds

Def. For a k-string TM M $(k \geq 1)$, M operates within space $f(n)$ if for every input x, the sum of the lengths of the k strings during the computation never exceeds $f(|x|)$.
(If M halts, we only need to consider the strings after halting, due to our conventions.)

- Note that f as above must satisfy $f(n) \geq n$ for all n.
- But we would also like to study how much extra space is needed besides input, and possibly output. (The extra space may be less than linear.)

Def. Let $k > 2$. A k-string TM M is called a <u>k-string TM with input and output</u> if:

a) string 1 is a "read only" input string.

b) string k is a "write only" output string.

c) the cursor never goes further to the right after reading the input string or after writing the output string (on the respective strings).

Def. For a k-string TM M with input and output, M <u>operates within space $f(n)$</u> if for every input x, the sum of the lengths of the strings 2 to $k-1$ (the "workspace") never exceeds $f(|x|)$.

Def. $L \in SPACE(f(n)) \iff$

there is k-string TM M $(k>2)$ with input and output which decides L and operates within space $f(n)$.

Example $L = \{\text{palindromes}\}$ belongs to $SPACE(\log n)$.

Constants don't matter for space complexity classes:

Thm 2.3 Let $L \in SPACE(f(n))$. Then, for every $\varepsilon > 0$, $L \in SPACE(\varepsilon f(n) + 2)$.

Nondeterministic TMs (NTMs)

Def. An NTM N decides L if for every x:

$x \in L \iff$ some computation with input x ends with "yes".

If in addition, for every x, N halts within $f(|x|)$ steps then N decides L within time $f(n)$.

Def. $L \in NTIME(f(n)) \iff$ some NTM N decides L within time $f(n)$.

P and NP

Def. $P = \bigcup_{k \in \mathbb{N}} TIME(n^k)$

$$NP = \bigcup_{k \in \mathbb{N}} NTIME(n^k).$$

Every TM is an NTM so $P \subseteq NP$.

Thm 2.6

$$NTIME(f(n)) \subseteq \bigcup_{c>1} TIME(c^{f(n)}).$$

Proof idea: "Dovetailing". Let $L \in NTIME(f(n))$ be decided by the NTM N in time $f(n)$. Construct a TM M with 3 strings. On one of them we simulate every every possible computation of N on the given input. Another string keeps track of which computations, coded as finite seq. of numbers $\leq d$ (d depends on M), have already been tried.

The hierarchy theorem

Without restrictions on f strange things can happen:

Thm 7.3 (The gap thm)

There is a recursive function $f: \mathbb{N} \to \mathbb{N}$ such that $TIME(f(n)) = TIME(2^{f(n)})$.

Proof idea: f is defined so that for every TM M and every input x, either M halts before $f(|x|)$ steps or after $2^{f(|x|)}$ steps.

Thus, we would like to restrict the functions under consideration, still retaining the interesting ones, such as

n^k, $\sqrt{n}$, $\log n$, 2^n, $n!$.

<u>Def.</u> f is a <u>proper (complexity) function</u> if there is a k-string TM M_f $(k > 2)$ with input and output such that for every input x:

(1) $M_f(x) = \sqcap^{f(|x|)}$ wher '$\sqcap$' is a "quasi-blank" symbol.

(2) M_f halts after $\mathcal{O}(|x| + f(|x|))$ steps.

(3) M_f operates within space $\mathcal{O}(f(n))$.

(4) $f(n+1) \geq f(n)$ for all $n \in \mathbb{N}$.

The above mentioned functions are proper (complexity) functions and if f and g are proper, then so are $f+g$, $f \cdot g$ and 2^f.

<u>Def.</u> For any func. f define

H_f = "the halting language of f" =

= $\{(M, x) : M$ accepts x within $f(|x|)$ steps $\}$

<u>Lemma 7.1</u> If f is a proper complexity function, then
$H_f \in TIME(f(n)^3)$.

<u>Proof.</u> Somewhat involved, using techniques from the proofs of previous results and Prop 7.1.

<u>Lemma 7.2</u> If f is a proper complexity function, then
$H_f \notin TIME(f(\lfloor \frac{n}{2} \rfloor))$.

<u>Proof idea</u> "Diagonalization" works here. If M_{H_f} decides H_f in time $f(\lfloor \frac{n}{2} \rfloor)$ then define

$$D_f(M) = \begin{cases} \text{no} & \text{if } M_{H_f}((M,M)) = \text{yes} \\ \text{yes} & \text{if } M_{H_f}((M,M)) = \text{no} \end{cases}$$

So on input M, D_f needs the same time as M_{H_f} on input (M,M), that is, time $f(\lfloor \frac{2n+1}{2} \rfloor)$ if M has length n.

By considering input D_f for D_f we get a contradiction:

$D_f(D_f) = \text{yes} \Longrightarrow$

$M_{H_f}((D_f, D_f)) = \text{no} \Longrightarrow$

D_f does <u>not accept</u> D_f within $f(|D_f|)$ steps

$\Longrightarrow D_f(D_f) = \text{no}$ because D_f <u>always</u> stops and says "yes" or "no" within $f(n)$ steps. (Other direction is analogous.)

By combining the previous two lemmas we get:

Thm 7.1 (Time hierarchy thm)

If f is a proper complexity function such that $f(n) \geq n$ for all $n \in \mathbb{N}$, then

$$TIME(f(n)) \subsetneq TIME(f(2n+1)^3).$$

proper

Corollary $P \subset EXP$, where

$$EXP = \bigcup_{k \in \mathbb{N}} TIME(2^{n^k}).$$

Thm 7.2 (Space hierarchy thm)

If f is a proper complexity function, then $SPACE(f(n)) \subset SPACE(f(n) \log f(n))$.