Bullet Notes on Computational Complexity

Albert Atserias,
Universitat Politècnica de Catalunya

MIRI, MAMME, MPAL
February-June, 2020.
Last modified: February 22, 2020.

1 Day 1

1.1 Model of computation

- A k-tape Turing machine (TM) is a 5-tuple $M = (Q, \Gamma, \delta, q_0, q_H)$ where:
- 1) Q is a finite set of *states*,
- 2) Γ is a finite set of *symbols* (the tape alphabet) containing the *blank* symbol $\Box$,
- 3) $\delta : Q \times \Gamma^k \to Q \times \Gamma^k \times \{\mathrm{L}, \mathrm{N}, \mathrm{R}\}^k$ is the *transition function*,
- 4) q_0 and q_H are two special states called *initial* and *halting* states, respectively.

- A *configuration* of M is $k+1$-tuple $c = (q, w_1, \ldots, w_k)$, where:
- 1) q is a state,
- 2) $w_1, \ldots, w_k$ are words from $\Gamma^*\$\Gamma^+$, where $\$ \not\in \Gamma$.
- $w_i = u\$av$ means that the tape contents is $\cdots \Box\Box uav \Box\Box \cdots$, and the head scans a.
- The *initial configuration* of M on input $x \in \Gamma^*$ is $(q, \$x\Box, \$\Box, \ldots, \$\Box)$.
- A configuration is called *halting* if $q = q_H$.

- Let c_1 and c_2 be configurations of M.
- We write $c_1 \vdash_M c_2$ if c_2 is the immediate successor of c_1 in the computation of M.
- We write $c_1 \vdash_M^t c_2$ if t steps of computation of M lead from c_1 to c_2.
- We write $c_1 \vdash_M^* c_2$ if $c_1 \vdash_M^t c_2$ for some $t \geq 0$.

- Example: If $\delta(q, 1) = (q', 0, R)$, then:
- $(q, 010\$11\Box 0) \vdash_M (q', 0100\$1\Box 0)$.
- $(q, 010\$11\Box 0) \not\vdash_M (q', 010\$01\Box 0)$.

- $(q, 010\$1) \vdash_M (q', 0100\$\square)$ [recall: $\cdots\square\square0101\square\square\cdots$].
- Example: If $\delta(q, 1) = (q', 0, L)$, then:
- $(q, \$100) \vdash_M (q', \$\square000)$ [again, recall: $\cdots\square\square100\square\square\cdots$]

- The *computation* of M on input $x \in \Gamma^*$, denoted $\mathrm{comp}_M(x)$, is:
- The unique sequence of configurations $s = (c_0, c_1, \ldots)$ such that:
- 1) c_0 is the initial configuration of M on input x,
- 2) $c_i \vdash_M c_{i+1}$ for each $i \in \{0, \ldots, |s| - 1\}$ or each $i \geq 0$ if s is infinite,
- 3) either s is infinite and no configuration in it is halting,
- 4) or s is finite and only the last configuration in it is halting.
- If $\mathrm{comp}_M(x)$ is finite, we call it a halting computation.
- If $\mathrm{comp}_M(x)$ is halting and $(q, w_1, \ldots, w_k)$ is the last configuration, then:
- The *output* is the word between \$ and the first $\square$ (both excluded) in $w_k\square$.
- We write $M(x) = y$ to mean that the computation halts *and* outputs y.

- The space of a configuration $c = (q, w_1, \ldots, w_k)$ is $|w_2| + \cdots + |w_k|$.
- We denote it $\mathrm{space}(c)$.
- Note that input tape doesn't contribute to space.
- This makes sense only if the input tape is read-only;
- I.e., $\delta(q, a_1, \ldots, a_k) = (q', a'_1, \ldots, a'_k, m_1, \ldots, m_k)$ requires $a'_1 = a_1$.

- The time of a computation $(c_0, c_1, \ldots)$ is its length; infinite if not halting.
- The space of a computation $(c_0, c_1, \ldots)$ is $\max\{\mathrm{space}(c_i) : i = 0, 1, \ldots\}$, or infinity if unbounded.

1.2 Deciding languages, computing functions

- Let $M = (Q, \Gamma, \delta, q_0, q_H)$ be a TM.
- Let $\Sigma \subseteq \Gamma$ be a finite alphabet.
- Let $L \subseteq \Sigma^*$ be a language and let $F : \Sigma^* \to \Sigma^*$ be a function.
- M computes F if $M(x) = F(x)$ for every $x \in \Sigma^*$.
- M decides L if $M(x) = \chi_L(x)$ for every $x \in \Sigma^*$, i.e.,
- 1) if $x \in L$ then $M(x) = 1$,
- 2) if $x \notin L$, then $M(x) = 0$.

1.3 Encodings

- For a finite object x, we use $\langle x\rangle$ to denote a fixed efficient binary encoding of x.
- Binary means: the encoding $\langle x\rangle$ of x is a word in $\{0,1\}^*$.
- Efficient means: length $|\langle x\rangle|$ of the encoding of x should not be inflated artificially.

- Strings: $\langle x\rangle = h_\Sigma(x)$, if $x \in \Sigma^*$ where $\Sigma = \{a_0 < \cdots < a_{|\Sigma|-1}\}$, and $h_\Sigma(a_i) = \mathrm{bin}(i)$.
- Pairs: $\langle x, y\rangle := 1^{|\langle x\rangle|}0\langle x\rangle\langle y\rangle$.
- Lists: $\langle x_1, \ldots, x_\ell\rangle := \langle x_1, \langle x_2, \ldots, x_\ell\rangle\rangle$ if $\ell \geq 1$, and $\langle\rangle := \lambda$.
- Naturals: $\langle n\rangle :=$ binary representation of n without unecessary leading zeros, for $n \in \mathbb{N}$.
- Integers: $\langle z\rangle := \langle b, n\rangle$, for $z = (-1)^b n \in \mathbb{Z}$ where $b \in \{0,1\}$ and $n \in \mathbb{N}$.
- Rationals: $\langle r\rangle := \langle b, p, q\rangle$, for $r = (-1)^b p/q \in \mathbb{Q}$ where $p, q \in \mathbb{N}$.
- Matrices: $\langle M\rangle := \langle M_{1,1}, \ldots, M_{1,n}, \ldots, M_{m,1}, \ldots, M_{m,n}\rangle$, for $M = (M_{i,j} : i \in [m], j \in [n])$.
- Graphs: $\langle A(G)\rangle$, for $G = (V, E)$ with $V = \{1, \ldots, n\}$ and adjacency matrix $A(G)$.
- ...

- Encoding of a Turing machine $M = (Q, \Gamma, \delta, q_0, q_H)$:
- $\langle M\rangle := \langle r, s, \langle p_1, a_1, q_1, b_1, m_1\rangle, \ldots, \langle p_t, a_t, q_t, b_t, m_t\rangle\rangle$ where:
- 1) $r = |Q| \geq 2$ and $s = |\Gamma| \geq 1$, and
- 2) the $(p_i, a_i, q_i, b_i, m_i)$'s enumerate all quintuples $\delta(p_i, a_i) = (q_i, b_i, m_i)$.
- Conventions:
- $Q = \{1, \ldots, r\}$ and $\Gamma = \{1, \ldots, s\}$,
- $q_0 = 1$ and $q_H = r$,
- $\square = 1$.
- Number $k \geq 1$ of tapes is readable from the quintuples p_i, a_i, q_i, b_i, m_i.

2 Day 2

2.1 Asymptotic growth rates

- Let $f : \mathbb{N} \to \mathbb{R}^+$ and $g : \mathbb{N} \to \mathbb{R}^+$ be functions.

- $f = O(g)$: $\exists c > 0\ \exists n_0 \geq 0\ \forall n \geq n_0\ f(n) \leq cg(n)$.
- $f = \Omega(g)$: $\exists c > 0\ \exists n_0 \geq 0\ \forall n \geq n_0\ f(n) \geq cg(n)$.
- $f = \Theta(g)$: $f = O(g)$ and $f = \Omega(g)$.

- $f = o(g)$: $\forall c > 0\ \exists n_0 \geq 0\ \forall n \geq n_0\ f(n) \leq cg(n)$.
- $f = \omega(g)$: $\forall c > 0\ \exists n_0 \geq 0\ \forall n \geq n_0\ f(n) \geq cg(n)$.

- $\lim_{n\to\infty} f(n)/g(n) = 0$ implies $f = o(g)$.
- $\lim_{n\to\infty} f(n)/g(n) \in (0, +\infty)$ implies $f = \Theta(g)$.
- $\lim_{n\to\infty} f(n)/g(n) = +\infty$ implies $f = \omega(g)$.

- Constant: $\Theta(1)$.
- Logarithmic: $\Theta(\log n)$.
- Polylogarithmic: $\Theta((\log n)^c)$ for some $c > 0$.
- Linear: $\Theta(n)$.
- Quasilinear: $\Theta(n \log n)$.
- Quadratic: $\Theta(n^2)$.
- Polynomial: $\Theta(n^c)$ for some $c > 0$.
- Quasipolynomial: $\Theta(n^{(\log n)^c})$ for some $c > 0$.
- Linear exponential: $\Theta(2^{cn})$ for some $c > 0$.
- Exponential: $\Theta(2^{n^c})$ for some $c > 0$.
- Doubly exponential: $\Theta(2^{2^{cn}})$ for some $c > 0$.
- ...

2.2 Running time and space

- Let Σ be a finite alphabet.
- Let M be a k-tape TM with $\Sigma \subseteq \Gamma$ with $k \geq 2$ and read-only input tape.
- Let $x \in \Sigma^*$ be an input.
- Let $\mathrm{comp}_M(x) = (c_0, c_1, \ldots, c_t)$.
- Time: $\mathrm{T}_M(x) = t$.
- Space: $\mathrm{S}_M(x) = \max\{\mathrm{space}(c_i) : i = 1, \ldots, t\}$.
- Worst-case time: $\mathrm{t}_M(x) = \max\{\mathrm{T}_M(x) : x \in \Sigma^n\}$.
- Worst-case space: $\mathrm{s}_M(x) = \max\{\mathrm{S}_M(x) : x \in \Sigma^n\}$.

- $\mathrm{TIME}(f) = \{L :$ there exists M that decides L and $t_M = O(f)\}$.
- $\mathrm{SPACE}(f) = \{L :$ there exists M that decides L and $s_M = O(f)\}$.

- A function $f : \mathbb{N} \to \mathbb{N}$ is time-constructible if:
- 1) f is monotone non-decreasing, and $f(n) \geq n$ for every $n \in \mathbb{N}$,
- 2) there exists a TM M such that $M(1^n) = 1^{f(n)}$ for every $n \in \mathbb{N}$,
- 3) $t_M = O(f)$

- A function $f : \mathbb{N} \to \mathbb{N}$ is space-constructible if:
- 1) f is monotone non-decreasing, and $f(n) \geq n$ for every $n \in \mathbb{N}$,
- 2) there exists a TM M such that $M(1^n) = 1^{f(n)}$ for every $n \in \mathbb{N}$.
- 3) $s_M = O(f)$

- Examples: $f(n) = c$, $f(n) = \lceil \log_2(n) \rceil$, $f(n) = n$, $f(n) = n^2$, $f(n) = 2^n$.
- Fact: If f and g are time/space-constructible, then $f \circ g$ is time/space-constructible.

2.3 Universal Turing machine

- Theorem [Turing 1936, Universal Turing Machine]:
- There exists a TM U such that for every TM M with tape alphabet Γ and every $x \in \Gamma^*$:
- $U(\langle M, x \rangle) = \langle M(x) \rangle$.
- Moreover, there exists a constant c_M independent of x such that:
- $T_U(\langle M, x \rangle) \leq c_M T_M(x)^2 + c_M$,
- $S_U(\langle M, x \rangle) \leq c_M S_M(x) + c_M$.

- Proof idea: Just run it:
- 1. Given M and x in the input.
- 2. Let $c = \langle q_0, \$\square, \ldots, \$\square \rangle$; an encoding of the initial configuration of M (without input tape).
- 3. Let $c := \text{next}_M(c, x)$; this requires scanning M and x.
- 4. If c is not halting, go back to 3.
- 5. If c is halting, extract $M(x)$ from c and output it.
- After i steps of simulation we have $|c| = O(i)$.
- Computing $\text{next}_M(c, x)$ takes time $O(|c|)$ to move heads.
- Total time: $\sum_{i=1}^{T_M(x)} O(i) = O(T_M(x)^2)$.
- Total space: $O(S_M(x))$.

3 Day 3

3.1 Exercise

- Exercise: Show that 2^n is time/space constructible.
- Solution: Implement a binary counter; apply amortized analysis to verify $O(2^n)$ running time.

3.2 Time and Space Hierarchy Theorems

- Theorem [Time Hierarchy Theorem]:
- Let f and g be time-constructible functions.
- If $f^2 = o(g)$, then $\mathrm{TIME}(f) \subsetneq \mathrm{TIME}(g)$

- Proof:
- Let D be the following TM:
- 1. Given an input z,
- 2. Use the time-constructibility of g to write down $1^{g(|z|)}$ on a designated tape.
- 3. Use the designated tape as a shut-down clock; if it runs over time, halt and output 0.
- 4. Find first 0 in z; if z has no 0's, halt and output 0.
- 5. Let m and x be such that $z = 1^{|m|}0mx = \langle m, x\rangle$.
- 6. Run U on input $\langle m, \langle m, x\rangle\rangle$, i.e., $\langle m, z\rangle$.
- 7. If $U(\langle m, z\rangle) = 0$, halt and output 1.
- 8. If $U(\langle m, z\rangle) = w \neq 0$, halt and output 0.
- The running time of D is $O(g)$; in particular it always halts.
- Let L be the language that D decides, so $L \in \mathrm{TIME}(g)$.
- We claim that $L \notin \mathrm{TIME}(f)$.
- Proof of claim:
- Let M be a TM that runs in time $O(f)$.
- Say $t_M(n) \leq cf(n)$ for every $n \geq n_0$.
- Let c_M be such that $T_U(\langle M, x\rangle) \leq c_M T_M(x)^2 + c_M$ for every x.
- Let $m = \langle M\rangle$ and $\ell = |m|$; i.e., the length of the encoding of M.
- Let n be large enough so that steps 4,5,6,7,8 on $z = \langle m, 1^n\rangle$ take time at most $g(2\ell + 1 + n)$.
- Such an n exists by $f(n) \geq n$ and $t_M(n) \leq cf(n)$ for every $n \geq n_0$, and $f^2 = o(g)$.
- Then D on input $z = \langle m, 1^n\rangle$ is not shut-down by the clock.
- Then $D(z) \neq U(\langle m, z\rangle) = M(z)$.
- So M does not decide D. QED

- Theorem [Space Hierarchy Theorem]:
- Let f and g be time-constructible functions.
- If $f = o(g)$, then SPACE(f) $\subsetneq$ SPACE(g)

- Proof: Same proof using the more efficient space simulation given by U. QED

3.3 Basic Complexity Classes

- $\mathrm{P} = \bigcup_{c>0} \mathrm{TIME}(n^c)$.
- $\mathrm{EXP} = \bigcup_{c>0} \mathrm{TIME}(2^{n^c})$.
- $\mathrm{EEXP} = \bigcup_{c>0} \mathrm{TIME}(2^{2^{n^c}})$.
- Corollary: P $\subsetneq$ EXP $\subsetneq$ EEXP.

- $\mathrm{LOGSPACE} = \bigcup_{c>0} \mathrm{SPACE}(c \log n)$ (also denoted L).
- $\mathrm{PSPACE} = \bigcup_{c>0} \mathrm{SPACE}(n^c)$.
- $\mathrm{EXPSPACE} = \bigcup_{c>0} \mathrm{SPACE}(2^{n^c})$.
- Corollary: LOGSPACE $\subsetneq$ PSPACE $\subsetneq$ EXPSPACE.

- LOGSPACE $\subseteq$ P $\subseteq$ PSPACE $\subseteq$ EXP $\subseteq$ EXPSPACE $\subseteq$ EEXP.

3.4 Nondeterminism

- A non-deterministic Turing machine (NTM) is a 6-tuple $M = (Q, \Gamma, \delta_0, \delta_1, q_0, q_H)$, where:
- Q, Γ, q_0 and q_H are as in deterministic TMs.
- δ_0 and δ_1 are two transition functions.
- Equivalently: $\delta : \{0,1\} \times Q \times \Gamma^k \to Q \times \Gamma^k \times \{\mathrm{L}, \mathrm{N}, \mathrm{R}\}^k$.
- Operation: Computation paths; in t steps, up to 2^t such paths.
- For $y \in \{0,1\}^*$, let $\mathrm{comp}_M(x, y)$ be computation path of M on input x using δ_{y_i} at step i.
- $T_M(x) =$ length of the longest computation path of M on input x.
- $S_M(x) =$ space of the most spacious computation path of M on input x.
- Worst-case time: $t_M(n) = \max\{T_M(x) : |x| = n\}$.
- Worst-case space: $s_M(n) = \max\{S_M(x) : |x| = n\}$.

- M decides L if for every $x \in \Sigma^*$ we have:

- 1) if $x \in L$, then $M(x) = 1$ on some computation path,
- 2) if $x \notin L$, then $M(x) = 0$ on every computation path.

- $\text{NTIME}(f) = \{L :$ there exists M that decides L and $t_M = O(f)\}$.
- $\text{NSPACE}(f) = \{L :$ there exists M that decides L and $s_M = O(f)\}$.
- $\text{NP} = \bigcup_{c>0} \text{NTIME}(n^c)$.
- $\text{NEXP} = \bigcup_{c>0} \text{NTIME}(2^{n^c})$.

- Fact: $\text{NTIME}(f) \subseteq \text{SPACE}(f)$.
- Proof:
- Let $L \in \text{NTIME}(f)$ and let M be a NTM that decides L in time $O(f)$.
- Build a DTM as follows:
- 1. Given an input x,
- 2. For $t = 1, 2, 3, \ldots$ do:
- 3. For $y = (y_1, \ldots, y_t) \in \{0, 1\}^t$ do:
- 4. Run up to t steps of M on input x using δ_{y_i} at step i.
- 5. If computation has halted and output 1, halt and output 1.
- 6. If computation has not halted or not output 1, go to next y.
- 7. If computation has halted and output 0 on all $y \in \{0, 1\}^t$, halt and output 0.
- Step 4 is executed by reusing space.
- Space usage:
- $O(\log f(n))$ for storing t,
- $O(f(n))$ for storing y and for step 4,
- $O(1)$ for bookkeeping and control.
- Remark: If f is space constructible, step 2 can be replaced by:
- 2. Produce $1^{f(|x|)}$ from $1^{|x|}$, let $t = cf(|x|)$,
- Here c is the constant such that $t_M(n) \leq cf(n)$ for all $n \geq n_0$.
- QED

- Theorem [Savitch Theorem]:
- $\text{NSPACE}(f) \subseteq \text{SPACE}(f^2)$.
- Proof:
- Let $L \in \text{NTIME}(f)$ and let $M = (Q, \Gamma, \delta, q_0, q_H)$ be a NTM that decides L in space $O(f)$.
- Let n_0 and c be such that $s_M(n) \leq cf(n)$ for all $n \geq n_0$.

- Number of configurations on inputs of length $n \geq n_0$:
- At most $|Q| \times ((|\Gamma| + 1)^{cf(n)})^k$, where k is the number of tapes.
- Length of longest computation path on inputs of length $n \geq n_0$:
- At most $|Q| \times ((|\Gamma| + 1)^{cf(n)})^k \leq 2^{df(n)}$ for all $n \geq n_1$, for some constants $d > 0$ and $n_1 \geq n_0$,
- Define: $\mathrm{PATH}_M(c_1, c_2; t) =$ "c_2 is reachable from c_1 in at most 2^t steps".
- Then: $\mathrm{PATH}_M(c_1, c_2; t+1) =$ "$\exists c(\mathrm{PATH}_M(c_1, c; t) \wedge \mathrm{PATH}_M(c, c_2; t))$".
- Want: $\mathrm{PATH}_M(c_0, c_H; df(n))$, where:
- c_0: the initial configuration of M on input x,
- c_H: a (or even the) halting configuration of M that outputs 1.
- Machine for $\mathrm{PATH}_M(c_1, c_2; t)$:
- 1. If $t = 0$, check $c_1 = c_2$ or $c_1 \vdash_M c_2$, and output accordingly.
- 2. If $t \geq 1$, let $t' = t - 1$ and do:
- 3. For all $c \in \{0, 1\}^{df(n)}$ do:
- 4. Run $\mathrm{PATH}_M(c_1, c; t')$, let b_1 be the output,
- 5. Run $\mathrm{PATH}_M(c, c_2; t')$, let b_2 be the output,
- 6. Output $b_1 \wedge b_2$.
- Steps 4 and 5 are run reusing space.
- Total space: $O(f(n)) \times t$, which is $O(f(n)^2)$ if $t = \log_2(2^{df(n)})$.
- Remark: This proof assumes space constructibility of f; else, dovetail. QED

- P $\subseteq$ NP $\subseteq$ PSPACE $=$ NSPACE $\subseteq$ EXP $\subseteq$ NEXP $\subseteq$ EXPSPACE $=$ NEXPSPACE $\subseteq \cdots$