

Computational Complexity

September 6 2006

1 ad previous lesson

Example of f being not time-constructible:

$$f(i) := \max_{L, x \in L \Rightarrow |x| \leq i} [\# \text{ steps of } U \text{ to terminate on } x],$$

where U is the universal TM.

Facts: f is not recursive, $\text{DTIME}(f) = \text{REC}$. That's because every recursive language L' can be viewed as $L' = L_1 \cup L_2$, where all words of L_1 are of length at most i . Deciding whether $x \in L'$, $|x| = i$ might be performed by running U for L_1 , where it certainly terminates in $f(i)$ steps.

2 DTIME and DSPACE hierarchies

Theorem 2.1. (*biimmunity*) *There is a language $L \in \text{DTIME}[2^{cn}]$ s.t. any algorithm for deciding L uses $> 2^n$ steps on almost all inputs (up to finitely many). Such a language is called 2^n -biimmune.*

Proof. Recall the DTIME-gap theorem from the previous lesson: There is an $L \in \text{DTIME}[2^{cn}]$ s.t. any algorithm deciding L uses $> 2^n$ steps on infinitely many inputs. Similarly to the proof of this theorem let us take the set $U = \{\langle M, x \rangle\}$, where M denotes a TM, and x denotes an input (in binary). Let us denote the i -th elements in the lexicographically ordered sets of TMs, resp. inputs by M_i , resp. x_i (then $|x_i| \in O(\log i)$ and $|M_i| \in O(\log i)$). Let us define the languages L , and L' as follows:

$$\begin{aligned} L &:= \{ \langle M_i, x_s \rangle \mid \langle M_i, x_s \rangle \text{ rejects in } 2^{|x_s|} \text{ steps} \wedge \forall j < i \text{ s.t. } \langle M_j, x_s \rangle \text{ terminates in } \leq 2^{|x_s|} \text{ steps} \\ &\quad \exists r < s \text{ s.t. } \langle M_j, x_r \rangle \in L \cup L' \} \\ L' &:= \{ \langle M_i, x_s \rangle \mid \langle M_i, x_s \rangle \text{ accepts in } 2^{|x_s|} \text{ steps} \wedge \forall j < i \text{ s.t. } \langle M_j, x_s \rangle \text{ terminates in } \leq 2^{|x_s|} \text{ steps} \\ &\quad \exists r < s \text{ s.t. } \langle M_j, x_r \rangle \in L \cup L' \} \end{aligned}$$

Then L is the language we are looking for. Obviously, to check whether $\langle M, x \rangle$ belongs to $L \cup L'$ it takes $O(2^{|M|+|x|+|x|})$ steps (since we have to examine at most all machines $\leq M$ on all inputs $\leq x$ and for every machine and input we have to check what happens in at most $2^{|x|}$ steps. L

cannot contain infinitely many easy instances, since for every M_i that contains infinitely many easy instances, there is an x_{j_i} such that $\langle M_i, x_{j_i} \rangle$ rejects iff $\langle M_i, x_{j_i} \rangle \in L$. That can be proven by induction on i . \square

Theorem 2.2. (DSPACE-hierarchy) For all space-constructible f and any g s.t. $f \in o(g)$

$$\text{DSPACE}(f) \subsetneq \text{DSPACE}(g).$$

Proof.

$$L := \{\langle M, x \rangle \mid \langle M, \langle M, x \rangle \rangle \text{ doesn't accept using } g(|\langle M, x \rangle|) \text{ cells}\}$$

Obviously $L \in \text{DSPACE}(g)$. If $L \in \text{DSPACE}(f)$, then there's a machine M' that proves $\langle M', x \rangle \in L$ in $f(|\langle M', x \rangle|) \in o(g(|\langle M', x \rangle|))$ steps, that means if $\langle M', x \rangle \notin L$, which is a contradiction. \square

Definition 2.3. L is a class of languages that are decidable in log-time: $L := \text{DSPACE}[O(\log n)]$

Proposition 2.4.

$$\text{DSPACE}[s(n)] \subseteq \text{DTIME}[n^{O(1)}2^{O(s(n))}]$$

Proposition 2.5.

$$L \subseteq P \subseteq \text{PSPACE} \subseteq \text{EXP} \subseteq \text{EXPSpace} \subseteq \text{E}^2\text{XP} \dots$$

3 Non-uniform models of computation

Definition 3.1. PSIZE is a class of languages decidable by circuits of polynomial size:

$$\text{PSIZE} := \{L \subseteq \{0, 1\}^* \mid \exists C_i \text{ b.c.}, C_i \leq p(i), C_i \text{ decides } L \upharpoonright \{0, 1\}^i\}$$

Lemma 3.2. $P \subsetneq \text{PSIZE}$

Proof. 1. $P \subseteq \text{PSIZE}$

tableau method: simulate each one of polynomially many TM steps by a polynomial number of gates

2. $P \subsetneq \text{PSIZE}$

counting argument: $|\text{PSIZE}| = 2^{\aleph_0}$ whereas $|P| = \aleph_0$

constructively: PSIZE contains undecidable problems. Let L be a unary language, for which

$$0^i \in L \Leftrightarrow M_i(\emptyset) \downarrow.$$

Then $L \in \text{PSIZE}$ (one gate circuit to answer yes/no), but such family of circuits is non-uniform, and because of undecidability of this problem it's not possible to construct them by TM. \square

Conjecture $\text{EXP} \notin \text{PSIZE}$

Conjecture $\text{EXP} \notin \text{SIZE}[2^{o(n)}]$

Fact Every decision problem is in $\text{SIZE}[2^n]$.

Lemma 3.3. *There is an L s.t. $L \notin \text{SIZE}[o(2^n/n)]$.*

Proof. By counting. There are 2^{2^n} boolean formulas on n bits, and at most $2^{O(s \log(s+n))}$ circuits of size s (recursion), thus $s \in \Omega(O(2^n/n))$ to cover all formulas. \square

Example of a language without a small circuit:

for $n \in \mathbb{N}$
 for all f boolean formulas on n bits
 for all C circuits of size $2^n/n$ check whether C computes f ;
 pick f with the large minimal circuit, and for $|x| = n$ accept if $f(x) = 0$ and reject otherwise

Definition 3.4. *A TM with advice M' is defined as follows:*

$$M' = (M, y_1, \dots, y_i, \dots, y_i \in \{0, 1\}^*),$$

where M is a TM. M' accepts x iff $M(\langle x, y_{|x|} \rangle)$ accepts, and M' rejects x iff $M(\langle x, y_{|x|} \rangle)$ rejects.

Definition 3.5. P/f is a class of languages for which $L \in P/f$ iff it can be decided by a TM with advice in poly-time and $|y_i| \leq f(i)$.

Definition 3.6.

$$P/\text{poly} := \bigcup_{p \text{ polynomial}} P/p$$

Proposition 3.7. $P/\text{poly} = \text{PSIZE}$

Proof. 1. $P/\text{poly} \subseteq \text{PSIZE}$

Analogously to the proof of $P \subseteq \text{PSIZE}$, a TM M' with an advice y_i which decides instances of length i in polynomial time, can be polynomially simulated by a polynomial circuit C_i .

2. $P/\text{poly} \supseteq \text{PSIZE}$

Obviously, the circuit C_i of size $p(i)$ for deciding instances of length i can be viewed as a polynomial advice $y_i := C_i$. Then the TM M reads the advice that describes the circuit of polynomial size in i and evaluates it (polynomially in i). \square

4 Non-determinism

Unlike TM, a non-deterministic (one-tape) TM (NTM) uses *transition relation* instead of transition function, which is

$$\Delta(x, q) \subseteq \{(\tilde{x}, \tilde{q}, w \in \{\leftarrow, \downarrow, \rightarrow\})\},$$

where x, \tilde{x} are tape-symbols, q, \tilde{q} states, and w denotes the next move of the head on the tape. NTM accepts x , if there is some computational path that accepts x .

Definition 4.1.

$$\text{NTIME}(f) = \{L \mid x \in L \text{ is accepted by a NTM in } f(|x|) \text{ steps on all computational paths}\}$$

Definition 4.2.

$$\text{NP} = \bigcup_{p \text{ polynomial}} \text{NTIME}(p)$$

Proposition 4.3.

$$\text{L} \subseteq \text{NL} \subseteq \text{P} \subseteq \text{NP} \subseteq \text{PSPACE} \subseteq \text{NPSpace} \subseteq \text{EXP}$$