Aalto University
School of Science

CS-E4530 Computational Complexity Theory

Lecture 11: Hierarchy Theorems

Aalto University
School of Science
Department of Computer Science

Spring 2019

Agenda

- Time hierarchy theorem
- Space hierarchy theorem
- Consequences of hierarchy theorems

Lower Bounds?

- **We have argued that some problems *seem* intractable because they are complete for some complexity class**
  - NP-hard, PSPACE-hard, EXP-hard, . . .
- **However, we *have not* proven any unconditional resource lower bounds**
  - Only result of this type so far: undecidability
  - Is it possible e.g. that all decidable problems can be solved in polynomial time?

Recap: Time-constructible Functions

Definition (Time-constructible function)

Let $T\colon \mathbb{N} \to \mathbb{N}$ be a function. We say that T is *time-constructible* if $T(n) \geq n$ and there is a TM M that computes the function $x \mapsto \llcorner T(|x|) \lrcorner$ in time $T(n)$, where $\llcorner n \lrcorner$ denotes the binary representation of the number n.

Recap: Turing Machine Encoding

- **There is a mapping that maps each $\alpha \in \{0,1\}^*$ to a Turing machine M_α**

- **Mapping $\alpha \mapsto M_\alpha$ can be constructed to have the following properties**
  - Each TM is represented by *infinitely many strings*
  - Each string represents *some Turing machine*

Recap: Universal Simulation

Theorem

There is a TM $\mathcal{U}$ such that for every $\alpha, x \in \{0,1\}^*$,

- *if M_α halts on input x, then $\mathcal{U}\big((\alpha, x)\big) = M_\alpha(x)$, and*
- *if M_α does not halt on input x, then $\mathcal{U}$ does not halt on (α, x).*

Moreover, if M_α halts on input x in T steps using S space, then $\mathcal{U}$ halts on input (α, x) in $C \cdot T \log T$ steps using $C \cdot S$ space, where C is a constant that only depends on M_α.

- Universal simulation can be modified so that on input (α, x, t) the machine M_α is simulated on input x for t steps (t encoded in binary)

Time Hierarchy Theorem

Theorem

Let $f, g\colon \mathbb{N} \to \mathbb{N}$ be time-constructible functions satisfying $f(n) = o(g(n))$. Then

$$\mathsf{DTIME}(f(n)) \subsetneq \mathsf{DTIME}(g(n)\log g(n)) .$$

- **There are problems of *almost any possible* time complexity**
  - The factor $\log g(n)$ comes from the universal simulation
  - Quadratic simulation: $\mathsf{DTIME}(f(n)) \subsetneq \mathsf{DTIME}((g(n))^2)$
  - Linear simulation: $\mathsf{DTIME}(f(n)) \subsetneq \mathsf{DTIME}(g(n))$

Time Hierarchy Theorem: Proof

- **Rough idea:**
  - Define a function that has different value on some input than all functions computed by Turing machines running in time $o(g(n))$
  - This function can be computed in time $O(g(n) \log g(n))$
- **Define a computational problem L_g as follows:**
  - Let $\alpha \in \{0,1\}^*$ be the input
  - If machine M_α halts and outputs b on input α in time $g(|\alpha|)$, produce a different output (e.g. $1-b$)
  - If machine M_α does not halt on input α in time $g(|\alpha|)$, produce output 1
  - Put differently:

$$L_g = \{\alpha \in \{0,1\} : M_\alpha \text{ does not accept } \alpha \text{ in time } g(|\alpha|)\}.$$

Time Hierarchy Theorem: Proof

- ***First part***: $L_g \in \mathrm{DTIME}(g(n)\log g(n))$
- **On input** $\alpha \in \{0,1\}^*$**:**
  - Compute value $g(|\alpha|)$ in time $O(g(|\alpha|))$ time (g is time-constructible)
  - Run universal simulation for $g(|\alpha|)$ steps and decide output (takes $O(g(n)\log g(n))$ time)

Time Hierarchy Theorem: Proof

- ***Second part*:** $L_g \notin \text{DTIME}(f(n))$
- **Assume L_g is decided by M_g in time $c \cdot f(n)$**
  - Since $f(n) = o(g(n))$, there is some n_0 such that $c \cdot f(n) < g(n)$ for all $n \geq n_0$
  - By the properties of the Turing machine encoding, there is a string α such that $|\alpha| \geq n_0$ and $M_\alpha = M_g$
  - Does it hold that $\alpha \in L_g$?
    - $M_g(\alpha) = b \in \{0, 1\}$
    - Since $c \cdot f(|\alpha|) < g(|\alpha|)$, the machine M_g produces output b in $g(\alpha)$ steps, and by definition of L_g, the output must be $1 - b$
  - This is a *contradiction*

Nondeterministic Time Hierarchy

Theorem

Let $f, g\colon \mathbb{N} \to \mathbb{N}$ *be time-constructible functions satisfying* $f(n+1) = o(g(n))$. *Then*

$$\mathsf{NTIME}(f(n)) \subsetneq \mathsf{NTIME}(g(n)).$$

- Requires a somewhat different proof (omitted)

Recap: Space-constructible Functions

Definition (Space-constructible function)

Let $S\colon \mathbb{N} \to \mathbb{N}$ be a function. We say that S is *space-constructible* if there is a TM M that computes the function $x \mapsto \llcorner S(|x|) \lrcorner$ in space $O(S(n))$, where $\llcorner n \lrcorner$ denotes the binary representation of the number n.

Time Hierarchy Theorem

Theorem

Let $f, g : \mathbb{N} \to \mathbb{N}$ *be space-constructible functions satisfying* $f(n) = o(g(n))$. *Then*

$$\mathsf{SPACE}(f(n)) \subsetneq \mathsf{SPACE}(g(n)) .$$

- **Same proof as for the time hierarchy theorem**
  - No overhead from simulation in terms of space

Consequences of Hierarchy Theorems

- **Hierarchy theorems give separations between complexity classes**
  - P $\neq$ EXP
  - NP $\neq$ NEXP
  - L $\neq$ PSPACE
- **Hierarchy theorems give separations inside complexity classes**
  - DTIME$(n^k) \neq$ DTIME(n^{k+1}) for any $k \geq 1$

Lecture 11: Summary

- Time hierarchy theorem
- Space hierarchy theorem