Complexity Theory

Georg Moser

Institute of Computer Science @ UIBK

Summer 2008

Schedule

Time and Place

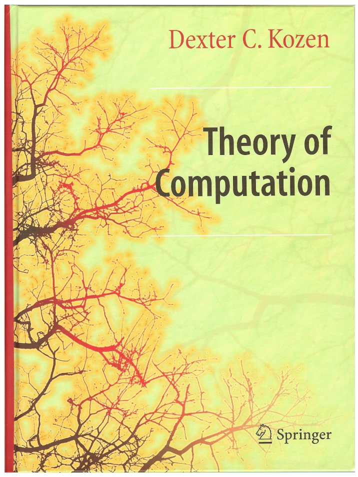
- Friday, 10:00-12:30, HS 10 (10 am sharp)

week 1	March 7	week 8	May 16
week 2	March 14	week 9	May 23
week 3	April 4	week 10	May 30
week 4	April 11	week 11	June 6
week 5	April 18	week 12	June 13
week 6	April 25	week 13	June 20
week 7	May 9	week 14	June 27
		first exam	July 4

Literature & Online Material

Literature

Theory of Computation, Dexter C. Kozen,
426 pages, Springer,
1st edition (March 23, 2006)
ISBN 1846282977



Online Material

Transparencies and homework will be available from IP starting with 138.232 after the lecture; exercises and solutions will be discussed during the lecture

Evaluation

Some are More Equal

master students	PhD students
very active participation	remarkable participation
do all the homework	scan the homework
solve all exercises	solve some interesting exercises
ace the exam	ace the exam

Exercises & Exam

- officially there are no exercises as this course is labelled VO
- however, without exercises you'll simply fail the exam
- I'll give weekly exercises which will be discussed in the lecture

Any protest?

Content

week 1	The Complexity of Computations
week 2	Time and Space Complexity Classes and Savitch's Theorem
week 3	Separation Results
week 4	The Immerman Szelepcsenyi Theorem
week 5	Logspace Computability
week 6	The Circuit Value Problem
week 7	Alternation
week 8	Problems Complete for PSPACE
week 9	The Polynomial Time Hierarchy
week 10	More on the Polynomial Time Hierarchy
week 11	Parallel Complexity
week 12	Relation of NC to Time-Space Classes
week 13	Probabilistic Complexity
week 14	Interactive Proofs

Overview and Goals

Kozen's book is based on a "*one-semester course for first-year graduate students in computer science at Cornell*"

Academic Ranking of World Universities

1	5	Massachusetts Inst Tech
2	2	Stanford Univ
3	26	Univ Illinois - Urbana Champaign
4	3	Univ California - Berkeley
5	21	Univ Michigan - Ann Arbor
		...
16	12	Cornell Univ
		...
-	320	Innsbruck Univ

☹

BUT

- a major in CS at Cornell comprises 4 years ...
- and this course sits in the 2nd year of the master program
- you have already heard all the basis courses for this course:
  - formal methods
  - algorithms and data structure
  - formal language theory
  - logic
  - computability theory

[...] we make use of discrete mathematical structures, including graphs, tress, and dags, O(.) and o(.) notation; finite automata, regular expressions, pushdown automata, and context-free languages; and Turing machines, computability, undecidability, and diagonalisation [...]

☺

Computational Complexity Theory

- define and study computational models and programming constructs
- understand their relative power and limitation
- classify computational problems in terms of their inherent complexity
- typically time or space complexity
- randomness, number of alternations, circuit size will also be of interest

prepared and studied by Church, Kleene, Post, Gödel, Turing, dots

Church's Thesis

all reasonable conceptions of computable functions capture our intuition of computable

Quantitative Church-Turing Thesis

tractable problems are those that are in the class of polytime decidable problems **P**

Consider $M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$ such that

$$Q = \{s, s_a, s_b, s_c, s_d, s_e, t, r\},\ \Sigma = \{0, 1\},\ \Gamma = \Sigma \cup \{\grave{0}, \acute{0}, \grave{1}, \acute{1} \vdash, \sqcup\}$$

with δ defined as:

$p \in Q$	$\gamma \in \Gamma$	$\delta_M(p, \gamma)$
s	$\vdash$	$(s_a, \vdash, \rightarrow)$
s_a	0	$(s_b, \grave{0}, \rightarrow)$
s_a	1	$(s_b, \grave{1}, \rightarrow)$
s_a	$\acute{0}$	$(s_e, \acute{0}, \leftarrow)$
s_a	$\acute{1}$	$(s_e, \acute{1}, \leftarrow)$
s_b	0	$(s_b, 0, \rightarrow)$
s_b	1	$(s_b, 1, \rightarrow)$
s_b	$\sqcup$	$(s_c, \sqcup, \leftarrow)$
s_b	$\acute{0}$	$(s_c, \acute{0}, \leftarrow)$
s_b	$\acute{1}$	$(s_c, \acute{1}, \leftarrow)$

$p \in Q$	$\gamma \in \Gamma$	$\delta_M(p, \gamma)$
s_c	0	$(s_d, \acute{0}, \leftarrow)$
s_c	1	$(s_d, \acute{1}, \leftarrow)$
s_c	$\grave{0}$	$(r, \grave{0}, -)$
s_c	$\grave{1}$	$(r, \grave{1}, -)$
s_d	0	$(s_d, 0, \leftarrow)$
s_d	1	$(s_d, 1, \leftarrow)$
s_d	$\grave{0}$	$(s_a, \grave{0}, \rightarrow)$
s_d	$\grave{1}$	$(s_a, \grave{1}, \rightarrow)$
s_e	$\grave{0}$	$(s_e, \grave{0}, \leftarrow)$
s_e	$\grave{1}$	$(s_e, \grave{1}, \leftarrow)$
s_e	$\vdash$	$(t, \vdash, -)$

One-Tape Turing Machine

A deterministic one-tape Turing machine is

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

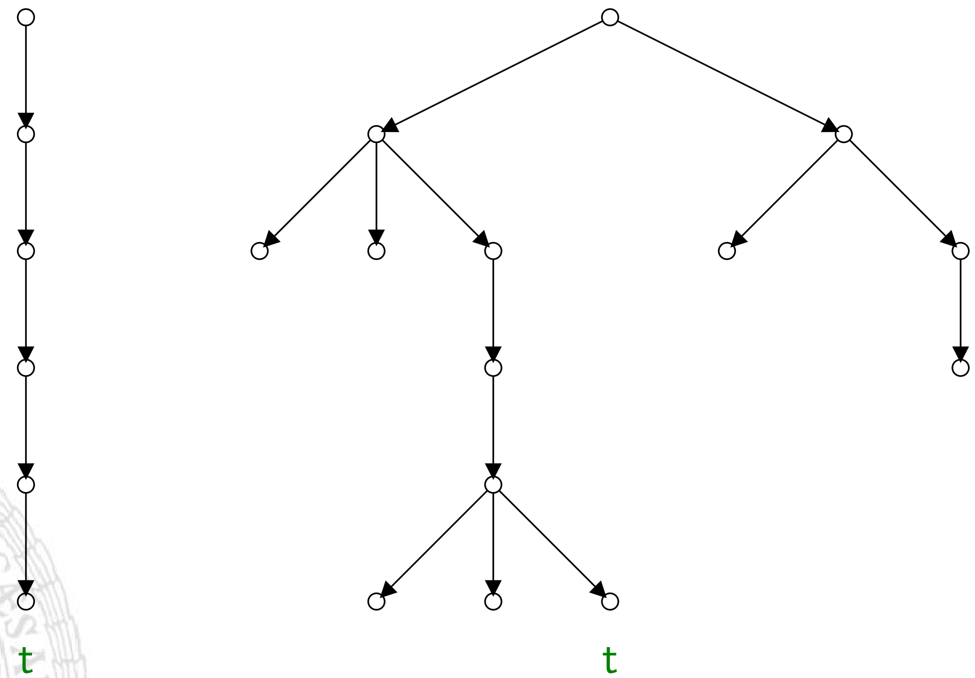
1. Q is a finite set of states
2. Σ is a finite input alphabet
3. Γ is a finite tape alphabet, $\Sigma \subseteq \Gamma$
4. $\sqcup \in \Gamma$ is the blank symbol
5. $\vdash \in \Gamma$ is the left endmarker
6. $\delta \colon Q \times \Gamma \rightarrow Q \times \Gamma \times \{\leftarrow, \rightarrow\}$ is the transition function
7. $s \in Q$ is the start state
8. $t \in Q$ is the accept state
9. $r \in Q$ is the reject state, $t \neq r$

$$\exists q\ \delta(p, \vdash) = (q, \vdash, \rightarrow) \qquad \exists c, d\ \delta(t, b) = (t, c, d)$$

$$\exists c', d',\ \delta(r, b) = (r, c', d')$$

Nondeterminism

Deterministic Nondeterministic



Nondeterministic One-Tape Turing Machine

A nondeterministic one-tape Turing machine is

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \Delta, s, t, r)$$

1. Q is a finite set of states
2. Σ is a finite input alphabet
3. Γ is a finite tape alphabet, $\Sigma \subseteq \Gamma$
4. $\sqcup \in \Gamma$ is the blank symbol
5. $\vdash \in \Gamma$ is the left endmarker
6. $\Delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{\leftarrow, \rightarrow\}$ is the transition relation
7. $s \in Q$ is the start state
8. $t \in Q$ is the accept state
9. $r \in Q$ is the reject state, $t \neq r$

$$\exists q\ \Delta(p, \vdash) = \{(q, \vdash, \rightarrow)\} \qquad \exists c, d\ \Delta(t, b) = \{(t, c, d)\}$$

$$\exists c', d',\ \Delta(r, b) = \{(r, c', d')\}$$

Theorem

Let $\Sigma = \{0, \mathbb{N}, \#\}$. The set of palindromes PAL $:= \{z \in \Sigma^* \mid z = \text{rev } z\}$ requires $\Omega(n^2)$ time on a one-tape TM

Proof

- $\text{PAL}_n := \{x \, \#^{\frac{n}{2}} \text{ rev } x \mid x \in \{0,1\}^{\frac{n}{4}}\}$
- all elements of PAL_n are of length n
- $\forall x \in \text{PAL}_n$, $\forall 0 \leqslant i \leqslant n$ let $c_i(x)$ denote the sequence of states

$$q_1, q_2, q_3, \ldots$$

$\in M$, if position i is passed (from left or from right) while scanning x

- $C(x) := \{c_i(x) \mid \frac{1}{4}n \leqslant i \leqslant \frac{3}{4}n\}$

Lemma

If $x, y \in PAL_n$ and $x \neq y$, then $C(x) \cap C(y) = \varnothing$ ■

Theorem

If M runs in $o(\log \log n)$ space, then M accepts a regular set

Fact

$\exists$ a non-regular set accepted in $O(\log \log n)$ space

$$\{\#b_k(0)\#b_k(1)\#b_k(2)\#\ldots\#b_k(2^k - 1)\# \mid k \geqslant 0\}$$

$b_k(n)$ is the k-bit (binary) representation of n

Proof

using similar crossing argument, but the following lemma

Lemma

If here is a fixed finite bound k on the amount of space used by M on accepted inputs, then $L(M)$ is a regular set ■