Complexity Theory (MTAT.07.004)

Autumn 2011

Peeter Laud & Bingsheng Zhang

class meets	Thu	16:15–17:45	(Liivi 2-404)
	Fri	14:15–15:45	(Liivi 2-405)

Books: S. Arora & B. Barak, Computational Complexity: A Modern Approach
C. Papadimitriou, Computational Complexity
M. Tombak, Keerukusteooria

webpage: http://www.cs.ut.ee/~peeter_l/teaching/keerukus11s

grading based on some homework, lecture scribing and final exam

Scribing

- In each lecture, a student will make detailed notes to be published on the course webpage.
  - With the current number of registered students, expect to be in charge of scribing during about two weeks of the semester.
- He/she prepares the notes in LaTeX and sends them to me before the next week's lectures.
- I will polish those notes and put them on the course webpage.
- See the course webpage for a template.

Models of computation

Computations

- Human computers: mid-17th — mid-20th century
  - Followed step-by-step instructions
- Notions of "computation" and "computability" formalized in mid-20th century.
- Turing machines, λ-calculus, cellular automata, Boolean circuits, random access machines, quantum circuits,...
- All those models are universal. Any computation performed in one of them can be modeled in another.
  - ...and with a similar* amount of computational effort
- How much resources does a computation need?

Resources

- Time
- Space
  - These two will be of interest in this course
- Program size
- Randomness
- Coherence
- ...

Turing machines — intuitive details

- k tapes, $k \geq 2$;
- first tape is the read-only input tape, other tapes are work tapes;
- tapes are infinite to the right only;
- the machine heads stay in place if they want to move left of the leftmost symbol;
- the alphabet contains bits 0 and 1, the blank symbol $\square$, the starting symbol $\triangleright$;
- the input string $x \in \{0,1\}^*$ is written on input tape as $\triangleright x \square\square \cdots$;
- initially, all non-input tapes contain $\triangleright \square\square \cdots$;
- initially, all heads are in the leftmost position;
- the answer y is written on the last tape as $\triangleright y \square\square \cdots$.

Turing machines

- A k-tape Turing Machine (TM) with input and output is a tuple $(\Gamma, Q, \delta, q_0, Q_F)$, where
  - Γ is the set of tape symbols;
    - Assume $\Box, \triangleright, 0, 1 \in \Gamma$
  - Q is the set of states;
  - $\delta : Q \times \Gamma^k \to Q \times \Gamma^{k-1} \times \mathsf{Move}^k$ is the transition function;
    - $\mathsf{Move} = \{-1, 0, 1\}$
  - $q_0 \in Q$ is the initial state;
  - $Q_F \subseteq Q$ is the set of final states.
- All sets above are finite.

TM Configurations

A configuration of a TM with k tapes, the tape symbol set Γ, and state set Q is

$$\langle q; w_1, \ldots, w_k; p_1, \ldots, p_k \rangle, \text{ where}$$

- $q \in Q$ is the current state of the TM;
- $w_i \in \Gamma^* \cdot \{\Box^\omega\}$ is the contents of the i-th tape.
  - w_i consists of a finite sequence of elements of Γ, followed by infinitely many $\Box$-s.
- $p_i \in \mathbb{N}$ is the position of the i-th head. Let leftmost position be 1.

Let $\mathcal{CONF}^k_{\Gamma,Q}$ be the set of all such configurations.

TM computations

A TM $M = (\Gamma, Q, \delta, q_0, Q_F)$ defines a relation (actually, a partial function) $\xrightarrow{M}$ on $\mathcal{CONF}^k_{\Gamma,Q}$.

$$\langle q; w_1, \ldots, w_k; p_1, \ldots, p_k \rangle \xrightarrow{M} \langle q'; w_1, w'_2, \ldots, w'_k; p'_1, \ldots, p'_k \rangle \text{ iff}$$

- $q \notin Q_F$
- $\gamma_i = w_i[p_i]$
- $(q'; \gamma'_2, \ldots, \gamma'_k; s_1, \ldots, s_k) = \delta(q; \gamma_1, \ldots, \gamma_k)$
- $w'_i = w_i[p_i \mapsto \gamma'_i]$
- $p'_i = \max(1, p_i + s_i)$

TM applied to a bit-string

- Let $M = (\Gamma, Q, \delta, q_0, Q_F)$.
- Let $x \in \{0,1\}^*$.
- Let $C_0 = \langle q_0; \triangleright \cdot x \cdot \Box^\omega, \triangleright \cdot \Box^\omega, \ldots, \triangleright \cdot \Box^\omega; 1, \ldots, 1\rangle$.
- Consider configurations $C_1, C_2, \ldots$, such that $C_{i-1} \overset{M}{\to} C_i$.
- If there exists $C_n = \langle q_n; w_1, \ldots, w_k; p_1, \ldots, p_k\rangle$ with $q_n \in Q_F$ then
  - we say that M stops on x in n steps in state q_n.
  - If also $w_k = \triangleright \cdot y \cdot \Box^\omega$ where $y \in \{0,1\}^*$ then we say that M outputs y on input x.
- If there is no such C_n, then M does not stop on x.

TM accepting a language

- A language L is any subset of $\{0,1\}^*$.
- Let $M = (\Gamma, Q, \delta, q_0, Q_F)$, where $Q_F = \{q_{\mathsf{acc}}, q_{\mathsf{rej}}\}$.
- If M
  - stops on all inputs $x \in \{0,1\}^*$;
  - stops in state q_{acc} iff $x \in L$

  then M accepts language L.

TM computing a function

- Consider functions f of type $\{0,1\}^* \to \{0,1\}^*$.
- Let $M = (\Gamma, Q, \delta, q_0, Q_F)$.
- If for all $x \in \{0,1\}^*$,
  - M stops;
  - M outputs y;
  - $y = f(x)$

  then M computes the function f.

Running time of a TM

- Let $T : \mathbb{N} \to \mathbb{N}$ and $f : \{0,1\}^* \to \{0,1\}^*$. The TM M computes f in time T, if it computes f, and for any $x \in \{0,1\}^*$, the machine M makes at most $T(|x|)$ steps.

- $T : \mathbb{N} \to \mathbb{N}$ is time constructible if $\forall n : T(n) \geq n$ and the function $x \mapsto \text{bit}(T(|x|))$ is computable in time $c \cdot T$ for some $c \in \mathbb{N}$.

  - $\text{bit}(n)$ is the representation of n as a binary string.

  Examples: n, $n \log n$, n^2, 2^n are time constructible.

- Non-time-constructible functions: try to encode the halting problem

Big-Oh notation

Let $f, g : \mathbb{N} \to \mathbb{N}$.

- $O(f), o(f), \Theta(f), \Omega(f), \omega(f)$ are sets of functions from $\mathbb{N}$ to $\mathbb{N}$.
- $g \in O(f)$ (or g is $O(f)$) if
  $\exists c \in \mathbb{R}_+ \ \exists n_0 \in \mathbb{N} \ \forall n \in \mathbb{N} : n \geq n_0 \implies g(n) \leq c \cdot f(n)$
  - $\mathbb{R}_+$ — all positive real numbers.
- $g \in o(f)$ if
  $\forall c \in \mathbb{R}_+ \ \exists n_0 \in \mathbb{N} \ \forall n \in \mathbb{N} : n \geq n_0 \implies g(n) \leq c \cdot f(n)$
- $g \in \Omega(f)$ if $f \in O(g)$.
- $g \in \omega(f)$ if $f \in o(g)$.
- $\Theta(f)$ is the intersection of $O(f)$ and $\Omega(f)$.

Reducing the size of the tape alphabet

Theorem. Let $M = (\Gamma, Q, \delta, q_0, Q_F)$ with k tapes accept a language / compute a function in time T. There exists a TM M' with $\max(k, 3)$ tapes and tape alphabet $\{\triangleright, \Box, 0, 1\}$ that accepts the same language / computes the same function in time $O(T)$.

Remark: Note that constant hidden in O may depend on M.

Multi-tape $\longrightarrow$ two-tape TM

Theorem. Let M with k tapes accept a language / compute a function in time T. There exists a TM M' with two tapes that accepts the same language / computes the same function in time $O(\lambda n.T(n)^2)$.

Actually, we can do better:

Theorem. Let M with k tapes accept a language / compute a function in time T. There exists a TM M' with three tapes that accepts the same language / computes the same function in time $O(\lambda n.T(n) \log T(n))$.

Two-way infinite $\longrightarrow$ one-way infinite tapes

Speeding up a TM

Theorem. Let a TM M compute a function / accept a language in time T. Then for each $c \in \mathbb{N}$ there exists a TM M' and constant c', such that M' computes the same function / accepts the same language in time $\lambda n.\frac{1}{c}T(n) + c'$.

Idea: Compute $6c$ steps of M "in hardware". This takes 6 steps on M'.

Turing machines as bit-strings

- A Turing machine $(\Gamma, Q, \delta, q_0, Q_F)$ can be represented as a bit-string.
  - State the number of tapes and the number of elements in Q and Γ. List the points of δ in some canonical order. Name q_0 and Q_F.
- For $\alpha \in \{0,1\}^*$, let M_α be the TM represented by it.
  - Let each bit-string represent some TM.

Universal Turing Machine

Theorem. There exists a five-tape TM $\mathcal{U}$ with tape alphabet $\{0, 1, \triangleright, \square\}$ and a function $C : \{0,1\}^* \to \mathbb{N}$, such that
for all $x, \alpha \in \{0,1\}^*$
if M_α on input x stops in t steps then

- $\mathcal{U}$ on input (α, x) stops in at most $C(\alpha) \cdot t \log t$ steps;
- the output of $\mathcal{U}$ on (α, x) equals the output of M_α on x.

- If M has two tapes then $\mathcal{U}$ stops in $C(\alpha) \cdot t$ steps.

Interpretation

- First convert M_α into a two-tape TM M'. Then reduce its alphabet to $\{0, 1, \triangleright, \square\}$ (add extra output tape).
- Then use the tapes as follows:
  1. The input tape of M'
  2. The work tape of M'
  3. The description of M'
  4. The current state of M'
  5. The output tape of M'

The complexity classes $\mathsf{DTIME}(f)$ and P

- Let $f : \mathbb{N} \to \mathbb{N}$
- The class $\mathsf{DTIME}(f) \subseteq 2^{\{0,1\}^*}$ is the set of all languages L, where
  - exists $g : \mathbb{N} \to \mathbb{N}$, such that
  - exists TM M that accepts L in time g, and
  - $g \in O(f)$.

$$\mathsf{P} = \bigcup_{c \in \mathbb{N}} \mathsf{DTIME}(\lambda n.n^c)$$

Random access machines (RAMs)

A RAM consists of two main parts:

- The register bank R.
  - Infinitely many registers, each capable of storing an integer.
- The program P:

1.	i_1
2.	i_2
3.	i_3
....	

- A RAM executes instructions until it jumps to the "final instruction" 0.
- Input — contents of register 0. Output — contents of register 0.

Instruction set of a RAM

- $T \leftarrow S$ and $T \leftarrow op(S)$ and $T \leftarrow S\ op\ S$
  - T is one of $R[n]$ or $R[R[n]]$. S is one of n or T.
- GOTO i and IF $R[n] > 0$ GOTO i.
- op comes from a fixed set of operations.
  - Must be careful in choosing those! Otherwise the machine can compute very fast.
  - Addition and subtraction are OK. Multiplication is not OK.

Simulating a TM on a RAM

Theorem. If $L \in \text{DTIME}(f)$, then exists a RAM M can accept L in time $O(f)$.

- Let M' be a k-tape TM that accepts L. Simulate L as follows:
- $R[1]$ encodes the state of M'.
- $R[2], \ldots, R[k+1]$ store the position of the read/write heads.
- $R[k+2], \ldots$ store the symbols on k tapes
  - One symbol per cell of R
  - k tapes are interleaved somehow
- $R[0]$ is used for arithmetic.
- The program of M is a translation of the transition function of M'.

Simulating a RAM on a TM

Theorem. If $L \subseteq \{0,1\}^*$ is accepted by a RAM M in time f then there exists a TM M' that accepts L in time $O(\lambda n.f(n)^3)$.

Consider a 7-tape TM.

- First tape: input string x (read-only).
- Second tape: contents of registers.
  - A sequence of elements of the form bit$i$: bit$R[i]$, ending with an end marker.
  - When updated, delete original pair, add new pair to the right, move end marker.
- Third tape: value of the program counter.

Simulating a RAM on a TM

- Fourth tape: the index of the register whose value is currently sought.
- Fifth and Sixth tapes: operands of the arithmetic operation.
- Fifth, sixth, seventh tape are used to perform arithmetic operations.

Instruction set of RAM may not allow the length of the contents of registers to grow too fast.