UMBC CMSC651, Automata Theory & Formal Languages, Fall 2010

Contact Information

Instructor: Prof. Richard Chang

Office:	ITE 326 [map]
Office Hours:	Tuesday & Thursday, 11:30am – 12:30pm
Telephone:	(410) 455-3093
E-mail:	chang@umbc.edu

Last Modified: 1 Sep 2010 22:44:42 EDT by Richard Chang

to Fall 2010 CMSC 651 Homepage

UMBC CMSC651, Automata Theory & Formal Languages, Fall 2010

Course Description

Textbook

Introduction to the Theory of Computation, second edition, Michael Sipser. Thompson Course Technology, ISBN 0-534-95097-3

Prerequisites

The undergraduate automata theory course (CMSC 451) is formally a prerequisite for this class. Where possible, this class will be self contained — i.e., students are not required to know many theorems that are not covered in class. However, it is important for the students in this class to be prepared to read and write mathematical proofs at a level that is consistent with having taken CMSC 451 or an equivalent course.

Objectives

- To understand some fundamental theorems in theoretical computer science.
- To develop the ability to write clear and convincing proofs.
- To build the background necessary for further studies in theoretical computer science.

Assignments & Grading

Your grade in this course will be based on 11 regular homework assignments, and three exams. Each homework assignment is worth 5 points and each exam is worth 15 points. The third exam will be during the time slot allocated for the final exam, but it will not be comprehensive. You are allowed to consult your classmates and others for the homework assignments (but you must write up the assignment yourself). The difference between assignments that have been copied and assignments that have been written up independently after the sharing of ideas is very obvious & especially in a small class. You are allowed to share ideas, but you are not allowed to copy. In general, homework assignments should be submitted when they are due, but reasonable allowances will be made for turning in regular assignments late.

Last Modified: 2 Sep 2010 00:20:36 EDT by Richard Chang
to Fall 2010 CMSC 651 Homepage

UMBC CMSC651, Automata Theory & Formal Languages, Fall 2010

Course Syllabus

The following schedule is a rough outline of the material to be covered during the semester. The chapters indicated are from *Introduction to the Theory of Computation,* by Michael Sipser.

I. Recursion/Computability Theory: 5 weeks.

HW1	assigned 09/02, due 09/09
HW2	assigned 09/09, due 09/16
HW3	assigned 09/16, due 09/23
HW4	assigned 09/23, due 09/30
Exam 1	in class 10/05

II. Basic Computational Complexity Theory: 4 weeks.

HW5	assigned 10/07, due 10/14
HW6	assigned 10/14, due 10/21
HW7	assigned 10/21, due 10/28
HW8	assigned 10/28, due 11/04
Exam 2	in class 11/09

III. Advanced Computational Complexity Theory: 4 weeks.

HW9	assigned 11/11, due 11/18
HW10	assigned 11/18, due 12/02
HW11	assigned 12/02, due 12/09
Exam 3	in class 12/21, 1pm – 3pm

Last Modified: 2 Sep 2010 00:18:46 EDT by Richard Chang

to Fall 2010 CMSC 651 Homepage

UMBC CMSC651, Automata Theory & Formal Languages, Fall 2010

Homework Assignments

Homework 1, Due Thursday, 09/09

1. Problem 3.14, page 161.
2. Problem 3.16, parts b, c & d, page 161.
3. Problem 3.19, page 162.
   Hint: you may assume that Problem 3.18 has been solved.

Homework 2, Due Thursday, 09/16

1. Show that the version of the Halting Problem defined below is undecidable.

   $\mathrm{K} = \{ i \mid M_i \text{ is a TM that halts on blank input} \}$.

   Note: do not use Rice's Theorem.
2. Problem 5.16, page 212.
3. Show that E_{TM} many-one reduces to FINITE, where

   $E_{\mathrm{TM}} = \{ \langle M \rangle \mid M \text{ is a TM such that } L(M) = \emptyset \}$

   and

   $\mathrm{FINITE} = \{ \langle M \rangle \mid M \text{ is a TM such that } L(M) \text{ is a finite set} \}$.

Homework 3, Due Thursday, 09/23

1. Problem ~~4.19~~ 4.18, page 184.
   Note: Carefully argue that a) the language of the machine M you constructed is actually separates A from B, and that b) M always halts even if other machines that M is simulating could run forever.

2. Express the following languages using first-order quantifiers (∃ and ∀ over natural numbers or strings) followed by a decidable predicate:

   COFINITE = { ⟨ M ⟩ | the complement of $L(M)$ is finite }

   SUBSET = { ⟨ M_1, M_2 ⟩ | $L(M_1) \subseteq L(M_2)$ }

   Use as few quantifiers as you can and place all of the quantifiers at the beginning. In particular, none of the quantifiers should appear after a negation.

3. Let £$_1$, £$_2$, £$_3$, ... be a list of linearly bounded automata (LBA) as defined in Section 5.1 of the textbook. We can encode LBAs in the same manner that TMs are encoded. We say that £$_i$ is *minimal* if for all j such that L(£$_i$) = L(£$_j$), ⟨ £$_i$ ⟩ < ⟨ £$_j$ ⟩. That is, £$_i$ is minimal if it has the lexicographically smallest encoding of all the LBAs that recognize the same language.

   Let

   MIN_LBA = { ⟨ £$_i$ ⟩ | £$_i$ is minimal }.

   Show that MIN_LBA is Turing recognizable.

Homework 4, Due Thursday, 09/30

1. Let MIN_LBA be as defined in Homework 3. Show that MIN_LBA is undecidable.

2. Show that INFINITE many-one reduces to ALL_{TM} where

   INFINITE = { ⟨ M ⟩ | M is a TM and $L(M)$ is an infinite set }

   and

   ALL_{TM} = { ⟨ M ⟩ | M is a TM and $L(M) = \Sigma^*$ }.

   Note: You must construct a computable function f such that for all TM's M

   $\langle M \rangle \in$ INFINITE if and only if $f(\langle M \rangle) \in \text{ALL}_{\text{TM}}$.

   Carefully argue that the f you constructed is indeed computable and that both directions of the "if and only if" hold.

3. Problem 6.22, page 243.

Homework 5, Due Thursday, 10/14

1. Show that the function $f(n) = n \log_2 n$ is space constructible. (See Definition 9.1 for the definition of "space constructible".)

2. Show that SPACE[n^5] is immune to SPACE[n^3]. That is, show that there is an infinite set $A \in$ SPACE[n^5] such that for all $B \subseteq A$, where $B \in$ SPACE[n^3], the set B must be finite.

3. Prove that any Turing recognizable set A can be written as

   $$A = C / D = \{ x \in \Sigma^* \mid w = xy \text{ for some } w \in C \text{ and } y \in D \}$$

   for some C and D in SPACE[log n].

Homework 6, Due Thursday, 10/21

1. Problem 9.13, page 362.

2. Show that P $\neq$ SPACE[n^3].
   Hint: think about *pad*(A, n^2) for some $A \in$ SPACE[n^6].

Homework 7, Due Thursday, 10/28

1. Show that the log-space transducers defined on page 324 of Sipser is transitive. I.e., show that if $A \leq_L B$ and $B \leq_L C$ then $A \leq_L C$.

2. The description of the PATH problem in Section 7.2 uses adjacency matrices to represent graphs on the input tape of a Turing machine. Show that a log-space transducer can convert a graph given as a list of edges into its adjacency matrix representation. Here, assume that '(', ',' and ')' are part of the input alphabet. Also, consider carefully how your transducer handles the case of large sparse matrices, where the input might look like

   (1,2), (2,8713470183741023847120).

   where the length of 8713470183741023847120 might be much larger than log n. (It might be $\sqrt{n}$, for example.)

   You may ignore isolated vertices (vertices without any edges attached). So, in the example above, you can generate an adjacency matrix for the graph

   (1,2), (2,3).

   I.e., if the vertices numbered between 2 and 8713470183741023847120 do not appear on the adjacency list, you can assume that they don't exist. However, you still have to deal with the fact that 8713470183741023847120 is a rather large number and might have length bigger than log n.

3. Argue that the Immerman-Szelepcsenyi Theorem extends to nonconstructible space bounds above log n. I.e., show that for all space bounds $s(n) \geq \log n$, NSPACE [$s(n)$] = coNSPACE [$s(n)$].

4. *Extra Credit:* Problem 8.23, page 331.
   Note: since the graph G might not be connected, it does not suffice to simply count the number of edges. If the undirected graph G does not have any cycles, then it is simply a collection of trees.

Homework 8, Due Thursday, 11/04

1. Problem 7.23, page 296.

2. Problem 7.36, page 298.
3. Problem 7.39, page 299.
4. *Extra Credit:* Problem 7.28, page 297.

Homework 9, Due Thursday, 11/18

1. Problem 10.12, page 412.
2. The definition of *circuit family* is given on page 354 (Definition 9.27). We say that a language L has polynomial-sized circuits if L is decided by a circuit family $C = (C_0, C_1, C_2, ...)$ where the size of C_n is bounded by a polynomial in n.

   Show that if a language L has polynomial-sized circuits, then there exists a sparse set S such that $L \in \mathrm{P}^S$.

   Hint: you can stuff the circuit C_n in the sparse set S, but how do you get it out?
3. Show that if there exists a sparse set S such that $\mathrm{coNP} \subseteq \mathrm{NP}^S$, then PH collapses to Σ^{P}_3.

   Hint: mimic Karp-Lipton-Sipser, but be careful how you check if the sparse set you guessed "works".

Homework 10, Due Thursday, 12/02

1. Problem 10.11, page 412.
2. We say that a language L is in the class PP if there exists a deterministic polynomial-time Turing machine M and a polynomial $p()$ such that

   $$x \in L \Rightarrow \mathrm{Prob}_r [\ M(x, r) \text{ accepts }] > \tfrac{1}{2}$$

   $$x \notin L \Rightarrow \mathrm{Prob}_r [\ M(x, r) \text{ accepts }] < \tfrac{1}{2}$$

   where the probability is taken over $r \in \{0,1\}^{p(n)}$ and $n = | x |$.

   Show that NP $\subseteq$ PP.
3. Let N be a nondeterministic Turing machine. We define the function $\#\mathrm{acc}_N(x)$ to be number of accepting paths in the computation of N on input x.

   We say that a function f is in the class #P if there exists a nondeterministic polynomial-time Turing machine N such that for all $x \in \Sigma^*$, $f(x) = \#\mathrm{acc}_N(x)$.

   Show that the class #P is closed under addition and multiplication. That is, for any two functions f and g in #P, the functions

   $$h_1(x) = f(x) + g(x)$$

and

$$h_2(x) = f(x) \cdot g(x)$$

must also be in #P.

Homework 11, Due Thursday, 12/09

Available in PDF: hw11.pdf.

Last Modified: 1 Dec 2010 23:47:26 EST by Richard Chang
to Fall 2010 CMSC 651 Homepage