

## Research paper

Each student will write a documented research paper of at least 600 words on a topic chosen by the student and approved by the instructor. Two or more sources should be from edited journals, peer-reviewed conference proceedings, or academic books.

Include references within the text of your paper and a bibliography at the end in standard bibliographic format, including author name and publication information.

Research should be in some area covered by the course, with meaningful references to the course's textbook, sources, or handouts. Reference to the course material will be an important aspect of the research paper.

In your paper,

- Define terms
- Place in course context
- Include reference to theoretical results (theorems, proofs).

For information on formatting and documenting research see, for example, [http://owl.english.purdue.edu/handouts/research/r\\_apa.html](http://owl.english.purdue.edu/handouts/research/r_apa.html) or [http://webster.comnet.edu/apa/apa\\_intro.htm](http://webster.comnet.edu/apa/apa_intro.htm).)

The work will be done in three stages, each time with instructor feedback.

### Proposal (due in first month)

Include:

- *Title:* of more than one word; like anything else in your proposal, you may change this later;
- *Initial abstract:* a paragraph or so about the topic, including some factual assertion about the topic and optionally a statement of what you hope to learn;
- *References:* short initial list of 2-3 sources in bibliographic format.

For links to papers with sample abstracts and sample bibliographic format, see [www.framingham.edu/~dkeil](http://www.framingham.edu/~dkeil).

### Preliminary draft (due in second month)

Draft should respond to instructor comments on proposal and should be at least 300 words.

### Grading criteria

- Definitions of terms
- Relevance to course material
- Quality of writing
- Quality of factual material
- Reference to theoretical results
- Use of sources

### Final draft (due in third month)

Final draft should respond to instructor comments about preliminary draft and should be worthy of posting at instructor's web site.

## Introductory assignment

### A. Background and self introduction

At the Discussion Board, Forum “Introduction,”

- (a) Please comment in some way on your background or expectations for the course.
- (b) State what most stayed in your mind in discussing this topic, and if you wish, the least clear concept that you encountered in this topic.

### B. Structural induction

Using the definitions in the *Background* slides related to graphs and trees, prove *by structural induction* the theorem among the following that corresponds to your classroom ID for the course:

0. For any full  $m$ -ary tree  $T$ ,  $|T| \bmod m = 1$ . A full  $m$ -ary tree is one in which every non-leaf node has exactly  $m$  children.
1. For any graph, the sum of the degrees of the vertices is even.
2. For any graph, the number of vertices with odd degree is even.

3. For any tree  $T = (V, E)$ ,  $|V| = |E| + 1$ , where  $|V|$  is the number of vertices and  $|E|$  is the number of edges.
4. The result of removing an edge from a tree is a non-tree.
5. There is exactly one path between any two vertices in a tree
6. The result of adding an edge to a tree is a non-tree. Do *not* use the theorem that a tree has  $|V| - 1$  edges.
7. The height of a complete binary tree with  $n$  vertices is  $\log_2 n$ .
8. Adding an edge to a cyclic graph cycle yields a cyclic graph.
9. Removing an edge from a acyclic graph yields a graph that is not connected
10. The sum of the degrees of all vertices in a graph is twice the number of edges.

### C. Assessment of assignment

On a scale of 0 to 1.0, please estimate the learning value of exercise B above.

## Topic 1 assignment (Diagonal proof)

Each student will solve one of each set of the following problems, corresponding to student classroom ID.

Please submit on paper, giving your name, the assignment number, and the problem number and letter, and restating the problem.

### A. Response to classroom discussion

- a. What most stayed in your mind in discussing this topic?
- b. For you, what was the *least* clear concept that you encountered in this topic?

### B. Countable and uncountable sets

According to set theory, which of these sets are finite (*F*)? Countably infinite (*C*)? Uncountable (*U*)? Explain your result briefly.

0. signed integers
1. quotients of two integers
2. integers from 0 to 10100000
3. reals from 4.0 to 5.0
4. reals of the form  $d.333333\dots$ , where  $d$  is a natural number
5. reals
6. the set of all sets of integers
7. functions on integers
8. bitstrings of finite length
9. Java programs
10. infinitely long lectures

### C. Proving sets countable

Describe how you would show that the following sets are countable:

0. binary numerals
1. natural numbers that are multiples of 5
2. decimal fractions
3. English-language sentences
4. powers of 2
5. Java programs
6. formulas in predicate logic
7. proofs in logic
8. finite bit vectors
9. prime numbers
10. rational numbers
11. pairs of natural numbers
12. cells in a three-dimensional grid

### D. Diagonal proof

Solve the problem in the Topic 1 study questions, Longer Answer, “1b. Proof of uncountability,” corresponding to your classroom ID.

### E. Coinduction

Consider the set of infinite sequences of inputs and outputs, when inputs are *pairs* of strings of symbols in the set DIGITS = {0 .. 9}, and outputs are strings of DIGITS. DIGITS = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.

0. Formally define the set of input pairs and the set of output strings.
1. How many different input pairs exist and why?
2. How many different output strings and why?
3. Describe  $A^\infty = \{(\alpha, \beta) \mid \alpha \in A, \beta \in A^\infty\}$  and  $A \in \mathbf{N} \times \{T, F\}$
4. Formally define the set of *infinite sequences of* input pairs and output strings.
5. What is the cardinality of the set of infinite sequences of input pairs, and why?
6. Formally define the set of *infinite sequences of* input/output pairs of natural numbers.
7. What is the cardinality of the set of infinite sequences of input pairs and output strings, and why?
8. Formally define the set of *infinite* bit streams.
9. Formally define the set of *infinite sequences of* pairs of bits.
10. Formally define the set of *infinite sequences of* pairs of symbols from alphabet  $\Sigma$ .

### F. Assessment of assignment

On a scale of 0 to 1.0, please estimate the learning values of exercises B to E above.

## Topic 2 assignment (DFAs and RLs)

Each student will solve one problem in each set A to E. The problem matches the student's one-digit classroom ID. Specify problem numbers and restate as appropriate. Use pencil or JFLAP for diagrams.

### A. Response to classroom discussion

- What most stayed in your mind in discussing this topic?
- For you, what was the *least* clear concept that you encountered in this topic?

### B. DFA construction

Construct a DFA that recognizes the language specified under "2a. Construct FA" in the Study Questions.

### C. Regular expressions

Write a regular expression for the language assigned for part A above, using DIGIT if necessary for  $\{0 \dots 9\}$ .

### D. Proof of correctness of DFAs

Prove by structural induction that your answer in part B is correct. You may refer either to the informal description you wrote for part B or to the regular expression you wrote for part C.

### E. NFA problems

- Give NFAs with the specified number of states, recognizing each of the following languages. For each
- Describe the process for deriving the corresponding DFA by subset construction.

- $\{w \mid w \text{ ends with } 00\}$  with 3 states
- $\{w \mid w \text{ contains the substring } 0101\}$ , with 5 states
- $\{w \mid w \text{ contains an even number of 0's or exactly two 1's}\}$ , with 6 states
- $0^*1^*0^*0$ , with 3 states
- $\{w \mid w\text{'s second-last symbol is } 0\}$
- Strings containing both 00 and 11, or neither 00 nor 11 }

- $(a|b)^*ba^*$
- $(a|b)^*ba^*$
- $(a|ab)^*ba^*$
- $(ab|ba)^*a^*$
- $(ab|ba)^*a^*$
- Inputs that start and end with 0 (accepts 01100, rejects 01101)

Sources: (#1-4) Sipser, 1997; Goddard, 2008.

### F. Pumping Lemma problems

Prove by the pumping lemma that the language among the following that corresponds to your student classroom ID is not regular:

- binary numerals that represent the squares of whole numbers
- binary numerals that represent  $2x^3 + 3x + 5$  for some whole number  $x$
- binary numerals that represent  $2x^2 - 1$  for some whole number  $x$
- binary numerals that represent  $x^4$  for some whole number  $x$
- binary numerals that represent some power of 3
- The complement of  $\{0^n1^n \mid n \geq 0\}$
- $\{w \mid w \text{ in } \{0,1\} \text{ is not a palindrome}\}$
- $\{a^x b^{2y} \mid y = x\}$
- $\{a^x b^y \mid y = x^2\}$
- $u1u^R$  where  $u^R$  is the string  $u$  reversed
- $\{0^m1^n \mid m \geq n\}$
- $x0x$  where  $x$  is a string
- $\{a^x b^y \mid x \neq y\}$
- $\{0,1\}^m 0^m$
- $\{w \mid n_0(w) \neq n_1(w)\}$

Source for some: Sipser, 1997; Linz, 2006.

### G. Assessment of assignment

On a scale of 0 to 1.0, please estimate the learning values of exercises B to F above.

## DFA coding problems (extra credit)

Solve the (a) or (b) version of the problem described below. *Document* your code to reflect how it implements a DFA.

Implement a lexical analyzer (tokenizer, scanner) in a higher-level language of your choice, based on a DFA  $M$  such that  $L(M) = L(E)$ , where  $E =$

- (a)  $(('(' | ')') | '+' | '-' | '*' | '/' | '>' | '<' | '=' | '==' | '>=' | '<=' | \{ '0', '1', \dots, '9' \}^*)^*$
- (b)  $(\{ '{' | '}' | \text{keyword} | \text{identifier} | '=' | '.' | ',' | '(' | ')' | \text{comment} \}^* \text{ where}$   
     $\text{identifier} = (\text{underscore} | \text{letter})$   
     $(\text{underscore} | \text{letter} | \text{digit})^*$   
     $\text{keyword} = \text{class} | \text{int} | \text{while} | \text{if}$   
     $\text{comment} = (/^*)(\text{non-}^*)(/^/)$

Thus, the regular language that  $M$  accepts should be the set of sequences of tokens used in arithmetic and relational expressions, with numeric literals, in languages like Java.

Note that  $L(M)$  includes strings that are not *valid* arithmetic expressions, e.g.,  $>+2*-45)($ .  $M$  should ignore white space.

Also, your program should output a linked list of token objects; each object should have two members: the string, e.g.,  $>=$  or  $32$  or  $($ ), etc., and the identifier of its lexical category. Use the following lexical categories, left-paren, right-paren, addition operator, multiplication operator, relational operator, numeric literal.

### Suggested approach

- The DFA's state will be represented by an integer variable. The state transition function is represented by a *switch* statement that assigns a new state value to the state variable depending on what the most recent input character is.
- The easiest way to develop and test is to read text files and display a "yes" or "no" result.
- In one of the states, a token has just been recognized. Make sure that when your DFA is in this state, a subprogram is called that adds an object to your linked list of token objects.
- Each lexical category may correspond to a DFA; some will be simple

## Topic 3 assignment (PDAs and CFLs)

Each student will solve one problem in each set A to E. The problem matches the student's one-digit classroom ID. Specify problem numbers and restate as appropriate. Use pencil or JFLAP for diagrams.

### A. Response to classroom discussion

- a. What most stayed in your mind in discussing this topic?
- b. For you, what was the *least* clear concept that you encountered in this topic?

### B. CFG-CFL

Solve the problem corresponding to your classroom ID in Study Questions, Topic 3, "3b. CFG."

### C. Linear CFG

Explaining your method, define a *right-linear* CF grammar for

0.  $(0 \mid 10)^*$
1.  $101^*$
2.  $1^* \mid (01)^*$
3.  $1(01)^*$
4.  $1^*0$
5.  $10(00)^*$
6.  $(11 \mid 10)^*$
7.  $(01)^*(10)^*$
8.  $0^*(10)^*$
9.  $(0 \mid 1)0^*$
10.  $(00 \mid 11)^*$

### D. Define CFG

Explaining your method, define a CF grammar for

0.  $\{ 0^m 1^n 2^p \mid m = n \text{ or } n = p \}$
1.  $\{ 0^m 1^n 2^p \mid m = n + p \}$
2. balanced parentheses  
(not only  $((()))$  but also  $()((()))$  etc)
3. the set of Reverse Polish Notation expressions
4.  $a^{2n}b^{3n}$
5.  $\{ x \mid n_0(x) = 2n_1(x) \}$
6. Formulas in propositional logic (identifiers,  $\neg, \wedge, \vee, \Rightarrow, '(', ')'$ )
7. Set expressions (identifiers,  $\{, \}, ', ', \cap, \cup, \subseteq, \subset, \in$ )
8. Regular expressions over  $\{0, 1, 2\}$  with operators for selection ( $|$ ) and iteration ( $*$ )
9.  $\{ x \mid \text{length}(x) = 3n_1(x) \}$
10.  $\{ x \mid n_0(x) = n_1(x) \}$  (equal numbers of 0's and 1's)
11.  $\{ x0x^R \mid x \in \{0,1\}^* \}$

### E. Parser problem (extra credit)

Write a top-down parser that accepts the language generated by the specified CF grammar, where all-caps expressions denote literal keywords and lower-case names denote lexical categories or nonterminals.

1.  $S \rightarrow \text{class-defn } S$   
 class-defn  $\rightarrow$  "class" identifier { decl-list }  
 decl-list  $\rightarrow$  type-spec identifier ( ) { stmt-list }  
 type-spec  $\rightarrow$  "int" | "double"
2.  $\text{expr} \rightarrow \text{expression op simple-expression}$   
 simple-expression  $\rightarrow$  num | (expr)  
 op  $\rightarrow$  \* | / | %
3.  $S \rightarrow \text{term op simple-expression}$   
 simple-expression  $\rightarrow$  num | (simple-expression)  
 op  $\rightarrow$  + | -

### F. Assessment of assignment

On a scale of 0 to 1.0, please estimate the learning values of exercises B to E above.

## Topic 4 assignment (Turing machines)

Each student will solve one problem in each set A to E. The problem matches the student's one-digit classroom ID. Specify problem numbers and restate as appropriate. Use pencil or JFLAP for diagrams.

### A. Response to classroom discussion

- a. What most stayed in your mind in discussing this topic?
- b. For you, what was the *least* clear concept that you encountered in this topic?

### B. TM construction

Solve the problem corresponding to your classroom ID in Study Questions, Topic 3, "4a, 4b. TM construction."

### C. Definitions, theorems, and proofs

For the numbered item corresponding to your classroom ID:

- (a) Formally define the class described;
  - (b) state the theorem formally in the language of predicate logic
  - (c) Present a proof sketch in your own words.
  - (d) State the proof method you used.
  - (e) Give sources where appropriate.
1. The class of non-deterministic TMs is equivalent to the class of deterministic TMs
  2. The TMs accept a superset of the RLs.
  3. The set of strings that consist of the same string concatenated to itself is TM decidable.
  4. The set of strings that consist of a series of 0s, followed by the same number of 1s, followed by the same number of 2s, is TM decidable.
  5. The two-dimensional TM is equivalent to the standard TM.
  6. There exists a universal TM.
  7. The TM model is not equivalent to the linear-bounded automaton model.
  8. There exists a language that is undecidable but is recognized by a TM.
  9. The two-stack automaton is equivalent to the TM.
  10. The multi-head TM is equivalent to the TM.
  11. The two-tape TM model is equivalent to the standard TM model.

### D. Decidability and undecidability

State formally the assertion below corresponding to your student number. Give a brief proof sketch or explanation. Give your sources where appropriate.

0. State the theorems by Cantor on the cardinality of reals, Gödel on incompleteness, and Turing on undecidability. What basic method do the proofs share?
1. Suppose it were proven that problem  $P_1$  is undecidable. How might this fact be used to show that problem  $P_2$  is undecidable?
2. The problem of whether a given TM halts on blank input is undecidable.
3. The problem of whether a given TM halts on all inputs is undecidable.
4. The problem of whether a given TM recognizes the language  $\emptyset$  is undecidable.
5. Reducibility can be used to show undecidability.
6. If the halting problem were decidable, then every recursively enumerable language would be decidable.
7. The problem of whether two given TMs accept the same language is undecidable.
8. All regular languages are TM decidable.
9. The problem of whether a string is in the language generated by a given context-free grammar is decidable.
10. The problem of whether a string is in the language generated by a given regular expression is decidable.

### E. Assessment of assignment

On a scale of 0 to 1.0, please estimate the learning values of exercises B to D above.

## Topic 5 assignment (RAMs and $\mu$ recursion)

### A. Response to classroom discussion

- a. What most stayed in your mind in discussing this topic?
- b. For you, what was the *least* clear concept that you encountered in this topic?

### B. Short-answer questions

See “Study questions,” topic 5, “Short and longer answer.” Answer two questions that correspond to your classroom ID.

### C. Programs in the $\mathcal{S}$ language

Write programs in the  $\mathcal{S}$  language (including macros defined in the source material, such as variable assignment, addition unconditional goto, and goto on less than 0) that do the following. Solve the problem corresponding to your classroom ID. Note that output is considered to be the value of  $y$  at termination of program. Use comments or a paragraph of text to explain the program code.

0. Input  $x$ , output  $2x$
1. Input  $x$  and output  $(x + 1)$
2. Input  $x$  and output  $(x - 1)$
3. Input  $x_1 \dots x_n$  and output  $x_2$
4. Input  $x$  and output 1 if  $x = 0$ , otherwise output 0.
5. Input  $x$  and output 1 if  $x > 0$ , otherwise output 0.
6. Input  $(x_1, x_2)$  and output  $(x_1 \dot{-} x_2)$  (monus).
7. Input  $(x_1, x_2)$ ; output 1 if  $x_1 > x_2$ , otherwise 0. OK to use monus.
8. Input  $(x_1, x_2)$ , output  $2x_1 + x_2$
9. Input  $(x_1, x_2)$ , output  $3x_1 \dot{-} x_2$
10. Input  $(x_1, x_2)$  and output  $(x_1 x_2)$  (multiplication).
11. Input  $(x_1, x_2)$  and output  $(x_1 \div x_2)$  (division).
12. Input  $(x_1, x_2)$  and output  $(x_1 \bmod x_2)$ .

### D. $\mu$ -recursive functions and semidecidability

Answer one of the following corresponding to your classroom ID.

0. Distinguish primitive recursion from  $\mu$ -recursion.
  1. What proof approach might show that TMs are equivalent to the  $\mathcal{S}$  language? Describe.
  2. What proof approach might show that TMs are equivalent to the  $\mu$ -recursive functions? Describe.
  3. What statement in the  $\mathcal{S}$  language enables looping, and how? How is looping done in recursive function theory?
  4. Explain the notion of *composition* in recursive function theory, and tell why we can say that the primitive-recursive functions are *closed under composition*.
  5. Show by construction, with respect to the RAM model of computation, that if  $f$  and  $g$  are computable functions, and  $(\forall x) (h(x) = g(f(x)))$ , then  $h$  is computable.
  6. Explain how to prove that when both a language and its complement are recursively enumerable, the language is Turing decidable.

For 7 to 11, use operations on functions to show that the following are  $\mu$ -recursive

7. multiplication
8. subtraction
9. division
10. exponentiation
11. Finding the smallest  $x > 5$ , such that  $x^3$  is odd

### E. Assessment of assignment

On a scale of 0 to 1.0, please estimate the learning values of exercises B to D above.

## Topic 6 assignment (Interaction)

### A. Response to classroom discussion

- a. What most stayed in your mind in discussing this topic?
- b. For you, what was the *least* clear concept that you encountered in this topic?

### B. Study questions

See “Study questions,” topic 6, “Short and longer answer.” Answer two short answer questions that correspond to your classroom ID.

### C. PTMs

(a) Use a transition system or  $f_M$  description to define a PTM as follows

(b) State why the computation is not modeled by any TM alone.

0. An adder
1. A command-line processor that performs file creation, copy, delete, and directory operations
2. An answering machine (see slide)
3. A microwave oven
4. A digital clock
5. A vending machine that accepts \$1 and \$2 amounts and outputs candy and drinks, respectively
6. A soda machine (see slide) that accepts a dollar bill even when quarters have been inserted.
7. A soda machine that allows pressing a “Coin Return” button
8. A soda machine that accepts dollars, quarters, , and dimes
9. A soda machine that charges \$1.25
10. An odometer

### D. Assessment of assignment

On a scale of 0 to 1.0, please estimate the learning values of exercises B to C above.

## Topic 7 assignment (Concurrency)

### A. Response to classroom discussion

- a. What most stayed in your mind in discussing this topic?
- b. For you, what was the *least* clear concept that you encountered in this topic?

### B. Forms of concurrency

*Define the following terms and categorize according to the type of concurrency most associated with the term:*

0. Java threads
1. cellular automata
2. neural networks
3. parallel algorithm
4. message passing
5. indirect interaction
6. multitasking
7. service
8. mission
9. multi-agent system
10. asynchrony

### B. Parallel models

*Define the following terms and relate to models of parallel or distributed computing:*

0. mesh architecture
1. PRAM
2. CREW
3. MIMD
4. CRCW
5. SIMD
6. cache coherency
7. shared memory
8. distributed memory
9. bulk synchronous parallel model
10. barrier synchronization

### C. Multi-stream interaction

*For the terms below, give a definition and discuss ways to model what the concept relates to.*

0. concurrency
1. distributed processing
2. multi-stream interaction
3. coordination
4. sensor networks
5. stigmergy
6. indirect interaction
7. multi-agent system
8. self-organization
9. autonomous agents
10. social networks

### D. Assessment of assignment

On a scale of 0 to 1.0, please estimate the learning values of exercises B to C above.