

David Keil  
63.460 Theory of computing  
Framingham State College  
Spring 2008

## **Introductory assignment**

At the 63.460 site at *Blackboard.com*, go to the Discussion Board, read the message in thread “Welcome” of forum 0, and respond to it by leaving your own message, which should comment in some way on your background or expectations of the course. (Questions are welcome to fulfill this assignment!)

[If submitting near end of semester: ] What were the most effective aspects of the presentation of this course for you?

David Keil  
63.460 Theory of computing  
Framingham State College  
Spring 2008

## 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. Sources should be peer-reviewed or from edited periodicals or 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.

### ***Proposal***

Include:

- title (which like anything else in your proposal, you may change 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);
- short initial list of 2-3 sources in bibliographic format (see syllabus).

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

### ***Preliminary draft***

Draft, to be submitted in April, should respond to instructor comments on proposal and should be at least 300 words.

### ***Final draft***

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

## Topic 1 group work (Cardinalities)

1. Each group will solve one of the following problems:
2. Let us consider a mobile robot that at each step of its existence, must decide whether to turn left or right 5 degrees, based on its percept and state at that instant. Define a robot's "behavior" as a set of infinite sequences of  $((input, state), output)$  pairs, observable for the robot, where *input* is what the robot senses at an instant in time, expressed as a finite string over a finite alphabet, *state* is a string, and *output* is in  $\{left, right\}$ . That is, inputs, states, and  $(input, state)$  pairs may be represented as bit sequences (i.e., natural numbers).
3. Use Cantor's diagonal proof method to show that the set of all possible robot behaviors is uncountable. Can every behavior be algorithmically generated by some computer program?
4. A predicate on natural numbers,  $p : \mathbf{N} \rightarrow \{0,1\}$  is a function that may be represented as an infinite sequence  $seq_p$  of values in  $\{0,1\}$ , where the  $j$ th bit in the sequence is the value of  $p(j)$ . For example, if  $p$  is the predicate *odd*, then the  $j$ th bit of the sequence  $seq_p$  is 1 if  $j$  is odd, 0 if  $j$  is even.  
A Java program *computes* a certain predicate if, on input of some number  $x$ , the program outputs a 1 if  $p(x) = 1$ , and outputs a 0 when  $p(x) = 0$ . Any program can be described as a finite sequence of symbols, e.g., bits.  
Use Cantor's proof method to show that some predicates are undecidable by any program; that is, for some predicate  $p$  there is no program that computes the function  $p$ .  
*Hint:* Show that no bijection exists between the set of Java programs and the set of predicates.
5. Show that the power set of strings over an alphabet  $\Sigma$  is uncountable (see proof in handout that the set of sets of natural numbers is uncountable).
6. In your own words, give Cantor's proof that rational numbers are countable.

## Topic 2 group work (DFAs and RLs)

### A. DFA construction

Construct a DFA that recognizes...

1. Inputs that start and end with 0 (accepts 01100, rejects 01101)
2. Inputs with an odd number of bits (accepts 01100, rejects 011101)
3. Inputs with exactly 2 '0's (accepts 0110, rejects 011001)
4. Binary numerals divisible by 4 (accepts 01100, rejects 01101)
5. Unary numerals divisible by 4 (accepts 1111, rejects 111)
6. Integers (accepts 237, rejects x237)
7. Real numbers (accepts 1.23, rejects 1.2.3)
8. Java identifiers (accepts t0, rejects 2t0)
9. Java relational operators (accepts >=, >, rejects =<)

For three groups,  
group 1 should do problems 1, 4, 7  
group 2: 2, 5, 8  
group 3: 3, 6, 9

Post solutions as replies to this post. List names of students who participated. Subject line should specify group # or problems done. Suggestion: Use JFLAP to draw DFA diagrams. Also OK to write solution as text, listing each edge as (s, a, t) where s is a state, a is a symbol, t is a state.

### B. Regular expressions

Write regular expressions for the languages assigned for group work "A. DFA construction".

### C. 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.

1.  $\{ w \mid w \text{ ends with } 00 \}$  with 3 states
2.  $\{ w \mid w \text{ contains the substring } 0101 \}$ , with 5 states
3.  $\{ w \mid w \text{ contains an even number of 0's or exactly two 1's} \}$ , with 6 states
4.  $0^*1^*0^*0$ , with 3 states
5.  $\{ w \mid w \text{'s second-last symbol is } 0 \}$
6.  $\{ w \mid w \text{ has both } 00 \text{ and } 11 \text{ or neither } 00 \text{ nor } 11 \text{ as substrings} \}$

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

### D. Coding assignment

See handout, "DFA coding problems." This assignment is subject to simplification and need not be submitted during the discussion of Topic 2.

### E. Pumping Lemma problems

Prove by the pumping lemma that the language among the following that corresponds to your group's number is not regular:

1.  $\{ 0^n 1^m 0^n \mid m, n \geq 0 \}$
2. The complement of  $\{ 0^n 1^n \mid n \geq 0 \}$
3.  $\{ 0^m 1^n \mid m \neq n \}$
4.  $\{ w \mid w \text{ in } \{0,1\}^* \text{ is not a palindrome} \}$

Source: Sipser, 1997.

## Topic 3 group work (PDAs and CFLs)

### A. Short answer

1. Explain the relationship between a language and a pushdown automaton.
2. Explain why any finite language must be context free.
3. Explain what is meant by  
 $(\forall L \subseteq \Sigma^*) \text{Regular}(L) \Rightarrow \text{Context-Free}(L)$ ,  
and why it is true.

### B. DFA vs PDA

1. What operations can a PDA perform that a DFA cannot?
2. List differences and similarities between the DFA and the PDA models of computation. Give an example of how a stack is needed to recognize a non-regular language such as the ones listed in the DFA group work or in the textbook.

### C. CFG-CFL

7. Write a derivation tree for  
 $S \rightarrow Xa \mid aX \mid SX$   
 $X \rightarrow aY \mid Ya$   
 $Y \rightarrow ab \mid ba \mid \lambda \mid YY$   
and each of the following strings (according to your group number):
  8. 1. abbb;
  9. 2. aaba;
  10. 3. abab;
  11. 4. baab;
  12. 5. baaab

### D. Define CFG

1. Define a context-free grammar to specify the language consisting of all bit strings that have twice as many 0's as 1's. Explain your design briefly (one sentence).
2. Define a CF grammar for:
  - (a)  $\{ x \mid n_0(x) = n_1(x) \}$  (equal 0's and 1's)
  - (b) Palindromes
  - (c) Balanced parentheses (not only  $((()))$  but also  $()(())$  etc)
3. RPN challenge:
  - (a) Write a CF grammar for the set of all Reverse Polish Notation expressions.
  - (b) Define a PDA that accepts RPN expressions.
  - (c) Is RPN a regular language? If so, write a regular expression for it; otherwise show that it is not regular.
4. Define a CFG for the language  $0^n 1^{2n}$ . Explain your design briefly (one sentence).
5. Define a CFG for the language  $\{xx \mid x \in \Sigma\}$ . Explain your design briefly (one sentence).

### E. PDA coding problem

This follows up DFA problem (a).

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

```
S → class-defn S
class-defn → CLASS identifier { decl-list }
decl-list → type-spec identifier ( ) { stmt-list }
type-spec → INT | DOUBLE
```

## Topic 4 group work (TMs)

### A. TM construction

Write (as a diagram or table) the definition of the following Turing machine corresponding to your group number. Document by describing the purpose of each loop. If you use JFLAP, include as test results JFLAP traces of a few test inputs.

1. add a binary numeral to a unary numeral (tally, i.e.,  $11=2$ ,  $111=3$ , etc.), yielding binary output. Use '+' to separate the two parts of the input. For input  $100+11$  ( $4+3$ ), output should be  $111$  ( $7$ ).
2. binary inverter
3. unary to binary converter
4. unary to unary adder
5. unary to unary subtracter
6. unary equality comparison (on input  $x_1, x_2$ , if  $x_1 = x_2$  output 1, else 0)
7. unary greater-than (on input  $x_1, x_2$ , if  $x_1 > x_2$  output 1, else 0)
8. logical OR for two 1-bit inputs
9. logical AND for two 1-bit inputs
10. logical OR for multiple-bit input
11. logical AND for multiple-bit input
12. count 0's in input, display as unary numeral
13. compress bit string to eliminate 0's
14. binary equality comparison (on input  $x_1, x_2$ , if  $x_1 = x_2$  output 1, else 0)

15. binary greater-than (on input  $x_1, x_2$ , if  $x_1 > x_2$  output 1, else 0)
16. search (find location of first 1 in a bit string, output its location in unary)
17. binary to unary converter

### B. Decidability, undecidability

1. Explain why any language recognized by a DFA must also be recognized by some TM.
2. Compare and contrast the diagonal proofs by Cantor (cardinality of reals is greater than that of natural numbers) and Turing (some problems are uncomputable). What basic proof method do they share?
3. Suppose it were proven that problem  $P_1$  is undecidable. How might this fact be used to show that problem  $P_2$  is undecidable?
4. Briefly explain why no TM can solve the halting problem
5. Briefly explain why no TM can solve the problem of whether a given TM halts on blank input
6. Briefly explain why no TM can solve the problem of whether a given TM halts on blank input
7. Briefly explain why no TM can solve the problem of whether a given TM recognizes the language  $\emptyset$
8. What TMs solve semidecidable problems

## Topic 5 group work (RAMs and $\mu$ recursion)

### A. Programs

Write programs in the  $\mathcal{S}$  language that do the following. Each group solves the problem corresponding to the group number. Post solutions as replies to this post. List students who participated fully in solving the problem.

1. takes input  $x$  and outputs 1 if  $x = 0$ , otherwise outputs 0.
2. takes input  $x$  and outputs 1 if  $x > 0$ , otherwise outputs 0.
3. takes inputs  $(x_1, x_2)$  and outputs 1 if  $x_1 \geq x_2$ , otherwise outputs 0.
4. takes two inputs  $(x_1, x_2)$  and outputs  $\max\{0, (x_1 - x_2)\}$ .
5. takes two inputs  $(x_1, x_2)$  and outputs  $(x_1 x_2)$  (multiplication).
6. takes two inputs  $(x_1, x_2)$  and outputs  $(x_1 / x_2)$  (division).

### B. $\mu$ -recursive functions

1. Use the computability of addition to show that multiplication is  $\mathcal{S}$ -computable.
2. Use the computability of decrementing to show that subtraction is  $\mathcal{S}$ -computable.
3. Use the computability of multiplication to show that finding the smallest  $x$  such that  $(x^3$  is odd) is  $\mathcal{S}$ -computable.
4. Distinguish primitive recursion from  $\mu$ -recursion.
5. Distinguish primitive recursion from composition.

## **Topic 6 group work (Interaction and concurrency)**

For each of the terms below, (a) give a definition, (b) state whether you consider it a form of *computation* (and why or why not), (c) give examples of models, and (d) distinguish from the other terms in the list

1. concurrency
2. parallelism
3. distributed processing
4. sequential interaction
5. multi-stream interaction