University of Rochester	Computation and Formal Systems (CSC173)

Midterm Exam

Amy Murphy

24 October 2002

Read before beginning: This exam has 9 questions, some with multiple parts. The point value of each (sub)question is indicated in parenthesis. The total number of points is 100. I suggest reading the whole exam before starting.

Write your name on every page of the exam.

Please write clearly. Illegible answers cannot be graded. *Write all answers in the space provided.* Partial credit will be given, but only if you show your work. Do not leave any question blank!

This is a closed book, closed note exam. The proctor has scrap paper available, but it will not be collected. If your answer does not fit in the space provided, most likely you are trying to write too much. Also note, if your answer does not fill the space provided, this does not mean you should write more! If you ramble incorrectly, points will be deducted even if the correct answer is buried somewhere in your rambling.

While I have tried to make the questions as clear unambiguous as possible, if you do not understand a question or you think you are missing a piece of information, state as clearly as possible any assumptions you need in order to continue. In the name of fairness, the proctor has been instructed not to answer any questions.

Good luck!

Problem Number	Points Earned
1	/37
2	/6
3	/6
4	/10
5	/9
6	/9
7	/9
8	/6
9	/5
Total	/100

1. (37) **Relations.** For all parts of this problem, use the relations Figure 1. For your convenience, this figure is on the last page of the exam, and if you wish, you may remove it from the pack (i.e., rip the last page off of the exam).

   (a) (4) Write an English description of the following query:

   $$\pi_{PN}\sigma_{O=UR \wedge PN \neq PN_2 \wedge LN \neq LN_2}(((\mathit{Projects} \bowtie_{L=LN} \mathit{People}) \bowtie \mathit{Work}) \bowtie_{MD=MD} (\mathit{Projects} \bowtie \mathit{Work}))$$

   NOTE: If the attribute specified for a join has different names in the two relations being joined (e.g., L and LN), assume the name of the first is taken(L in this case). If there is an attribute naming conflict after a join, assume that the name of the second attribute is subscripted with a 2. i.e. if after the join, there are two attributes named A, the first one keeps the name A while the second is renamed A_2.

   Find all projects led by UR, where the projct meets the same day as another project, and at least one person does not serve as a member on both projects.

   (b) (3) What is the result of evaluating the previous expression?

   Milan, Infotonics, Systems

   (c) (5) Push the selection as far down as possible. Show your work for partial credit.

   Only the $O = UR$ part of the selection can be pushed down, and it is pushed all the way to the People relation.

   $$\pi_{PN}\sigma_{PN \neq PN_2 \wedge LN \neq LN_2}(((\mathit{Projects} \bowtie_{L=LN} (\sigma_{O=UR}\mathit{People})) \bowtie \mathit{Work}) \bowtie_{MD=MD} (\mathit{Projects} \bowtie \mathit{Work}))$$

   (d) (5) Take your solution from the previous question and push the projection as far down as possible.

   $$\pi_{PN}\sigma_{PN \neq PN_2 \wedge LN \neq LN_2}\pi_{\{PN,PN_2,LN,LN_2\}}((\pi_{\{PN,MD\}}(\mathit{Projects} \bowtie_{L=LN} (\sigma_{O=UR}\mathit{People})) \bowtie \mathit{Work})$$
   $$\bowtie_{MD=MD} \pi_{\{MD,PN,LN\}}(\mathit{Projects} \bowtie \mathit{Work}))$$

   (e) (6) Write the algebraic expression for the following query: *Find all project names whose leader is from UR and whose leader is also a member of another project.*

   $$\pi_{PN}(\sigma_{O=UR}(\mathit{Projects} \bowtie_{L=LN} \mathit{People}) \bowtie_{L=LN} \mathit{Work})$$

   (f) (3) What is the result of the previous query?

   Milan, Systems, Interweave

   (g) (5) For each relation, which attributes, or sets of attributes can serve as a key? For full credit, you must name ALL possible keys and no extra ones. Base your answer only on the current contents of each relation.

   People: LN. Projects: PN or L. Work: PN and LN.

   The most common error here was to include e.g., PN and L as a possible key for Projects. This is not correct because a subset of PN and L is also a key. According to the definition, this disallows it as a key.

   (h) (6) What is the difference between the primary index and the secondary index? What is the *cost* associated with adding an index to a relation?

   The primary index determines the storage of the relation in memory, while the secondary index provides only another lookup mechanism for the relation. The only storage necessary *for the secondary index is the pointers to the original data. The cost of adding an index initially is the cost to insert all of the elements (or pointers to the elements) of the relation into the new data structure. The cost to maintain the index is the extra time to insert and delete elements of the relation.*

2. (6) **Query Optimization.** In class we discussed several techniques for optimizing queries. Briefly describe indexing, selection pushing, and projection pushing and why each is effective for reducing the work needed to evaluate a query.

   Indexing provides fast lookup based on a specific index of a field. Typically an index uses a hash table using the index as a key, allowing O(1) lookup time.

   Selection pushing moves the selection operation "lower" in a query, eliminating tuples (rows) from the relation which will be used "higher" in the query, making the query more efficient.

   Project pushing moves the projection operation "lower" in a query, eliminating attributes (columns) from the relation. Because these columns will not be used in the "higher" parts of the query, projection pushing saves the space needed to process the query.

3. (6) Briefly define an abstract data type and a data structure, showing the difference between the two, and giving one example of each.

   An ADT describes a collection of data in terms of what operations it supports and how the results of the operations are related to one another. e.g., list, relation, set

   A data structure determines how the data is organized in memory and, consequently how fast the various operations can be performed. e.g., hash table, linked list, binary search tree

4. (10) **C.** Given the following code:

```
#include "stdio.h"
struct info {
  int val;
  char *str;
};

void f1(struct info in) {
  in.val = 4;
  strcpy(in.str,"56789");
}

void f2(struct info *in) {
  char *ch = (char*)malloc(5*sizeof(char));
  strcpy(ch, "45678");
  in->val = 2;
  in->str = ch;
}


int main() {
  struct info i;
  i.str = (char*)malloc(5*sizeof(char));
  i.val = 0;
  strcpy(i.str, "01234");
  printf("val = %d  str = %s \n",i.val, i.str);
```

```
  f1(i);
  printf("val = %d  str = %s \n",i.val, i.str);
  f2(&i);
  printf("val = %d  str = %s \n",i.val, i.str);
  return 0;
}
```

(a) (5) What output is produced?

```
val = 2 str = 01234
val = 2 str = 56789
val = 4 str = 45678
```

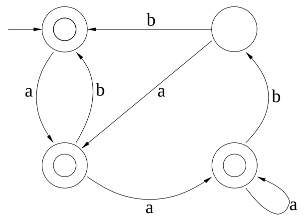
(b) (5) This code has a memory leak. Describe it, and a fix for it. If you wish, you may answer this question beside the code, indicating where your replacement code should be placed, and which, if any, lines of my code should be replaced.

The memory leak occurs because f2 mallocs memory which replaces the memory allocated in main, and the reference to the main-allocated memory is lost. One fix (although not the best) is to call free(i.str) before calling f2.

*Also, some of you mentioned that the amount of memory allocated is not sufficient for the 5 characters plus the newline. This is correct. I should have allocated 6*sizeof(char).*

5. (9) **Deterministic Finite Automata.**

(a) (4) Describe in English the language accepted by the following DFA. Do not simply describe how the machine works, give an intuitive description of the language.



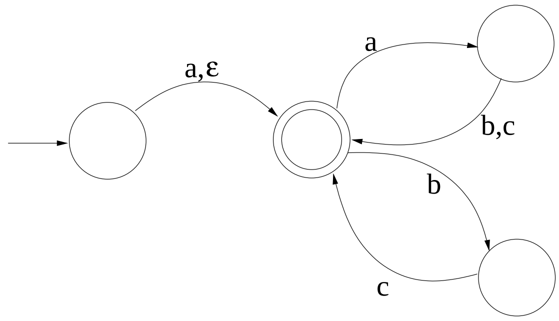
This problem had an error. There should have been an arrow from the start state back to the start state on the input b. With this correction, the atomata accepts strings which do not end with the substring aab.

(b) (5) Write a regular expression for the above DFA. Note: this may be tricker than it first appears.

$$(a|b)^*(aa|bb|ba|bab)|a|b|ab|\epsilon$$

6. (9) **Nondeterministic Finite Automata.** Given the following regular expression Draw an NDFA with 5 or fewer states which accepts only strings which are in the language. You may draw your NDFA with the assumption that if a transition does not exist for an input, the NDFA enters a dead state. You may use ϵ transitions in your NDFA. If your NDFA has more than 5 states, you can receive partial credit. Hint: do not do a mechanical transformation to an NDFA with epsilon transitions, followed by a reduction of the number of states.

$$(a|\epsilon)(ab|bc|ac)^*$$

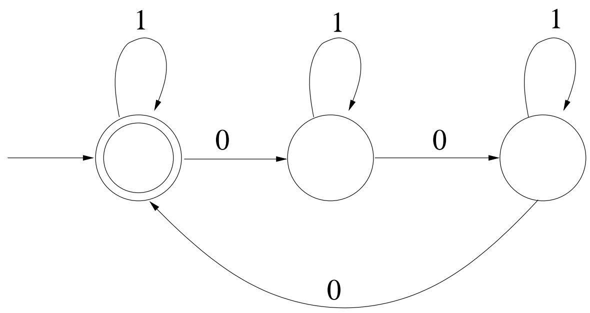


7. (9) **Regular Expressions.** Given the following regular expression

$$1^*(01^*01^*01^*)^*$$

   (a) (4) What is the plain English description of the language?
   Strings where the number of zeros is divisible by three.

   (b) (5) Draw a DFA which recognizes this language.



8. (6) **Equivalence.** We discussed several mechanical transformations in class, giving you a toolkit to do transformations among representations of a language. If you are given a regular expression, describe the sequence of transformations to produce a DFA, naming the technique used in the transformation.

   First, convert the RE into an NDFA with epsilon transitions using the constructions given in class and in the textbook. Next, convert the NDFA into a DFA using subset construction.

9. (5) **Practical use.** Name at least two places where regular expressions are used in common practice.

   Many answers were appropriate here. You could have mentioned: scanner of a compiler, searching a file for a substring with egrep, command line parameters to ls, etc.

People::	Last Name (LN)	Organization (O)
	Dwarkadas	UR
	Albonisi	UR
	Scott	UR
	Heinzelman	UR
	Murphy	UR
	Fauchet	UR
	Tai	IBM
	Lehman	IBM

Projects::	Project Name (PN)	Meeting Day (MD)	Leader (L)
	TSpaces	Fri	Lehman
	MILAN	Wed	Murphy
	Infotonics	Wed	Fauchet
	Systems	Fri	Dwarkadas
	Interweave	Mon	Scott

Work::	Project Name (PN)	Last Name (LN)
	TSpaces	Murphy
	TSpaces	Tai
	MILAN	Heinzelman
	Infotonics	Murphy
	Infotonics	Heinzelman
	Systems	Murphy
	Systems	Albonisi
	Systems	Scott
	Interweave	Dwarkadas

Figure 1: Toy database for question 1. Names in bold are the relation name. Attribute labels in parenthesis are used in the queries. Note: names and projects are mostly fictional.