0 Homework

Due: Wednesday, September 12, 2007.
Note: This review will be handled somewhat differently from all subsequent assignments. In particular:

- It will not be accepted late.
- You must attempt every problem
- Every point you earn on this assignment will count toward your homework score as extra credit points. (Please, review the grading policy described in the course information page for an explanation of how extra credit points are counted.)

1. (a) (5 pts) For each of these, give the resulting set by listing out all its elements:
   - $\{a, b, c\} \cap \{a, c, d, e\}$
     Solution
     $\{a, c\}$
   - $\{a, b, c\} \cup \{a, c, d, e\}$
     Solution
     $\{a, b, c, d, e\}$
   - $\{a, b, c\} - \{a, c, d, e\}$
     Solution
     $\{c\}$
   - $\{a, c, d, f\} - \{a, d, e\}$
     Solution
     $\{c, f\}$
   - $\{a, b, d\} \times \{a, b, c, d\}$
     Solution
     $\{(a, a), (a, b), (a, c), (a, d),$
     $(b, a), (b, b), (b, c), (b, d),$
     $(d, a), (d, b), (d, c), (d, d)\}$

   (b) (5 pts) Given any set S, the power of S, written $\mathcal{P}(S)$ or 2^S, is the set of all subsets of S. Write out $2^{\{a,b,c\}}$.
   Solution
   $\{\{\}, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$

2. (a) (5 pts) How many elements did you find in $2^{\{a,b,c\}}$? How many elements are there in $2^{\{a,b\}}$? How many elements are there in $2^{\{a,b,c,d\}}$? (It is not necessary to list all of them.) In general, if S is a finite set containing n elements (which we write as $|S| = n$), make a reasonable conjecture based on these examples for a formula for $|2^S|$ in terms of n.

   Solution

   $|2^{\{a,b\}}| = 4$

   $|2^{\{a,b,c\}}| = 8$

   $|2^{\{a,b,c,d\}}| = 16$

   For every set S such that $|S| = n$ $|2^S| = 2^n$.

   (b) (5 pts) Give a rigorous proof that your formula is correct for any $n \geq 0$. Hint: When creating a subset of S, for each element there is exactly one of two possibilities: it is either in this particular subset or it is not. Use this together with the product rule for counting the overall number of combinations when multiple options are possible. In particular, the product rule says that if there are three k_i options for selescting i^{th} item and each item may be selected independently of all other items, there are $k_1 k_2 ... k_n$ ways of selecting a combination of all n items.

   Solution

   Omitted

There are essentially two forms of notation we use to describe infinite sets in this class:

- using ellipses (i.e., ...); or
- using set-builder notation.

Here are two examples, described using ellipses:

- $\mathcal{N}$ = the set of all natural numbers = $\{0, 1, 2, 3, ...\}$; and
- $\mathcal{Z}$ = the set of all integers = $\{..., -3, -2, -1, 0, 1, 2, ...\}$.

Here is another example, which we define using both methods: The set of all natural numbers that are perfect squares is

$$\{0, 1, 4, 9, 16, 25, ...\} = \{n | n = m^2 \text{ for some } m \in \mathcal{N}\} = \{n^2 | n \in \mathcal{N}\}$$

Note that whenever a set is infinite, only set-builder notation gives a mathematically rigorous specification of that set. If a set is infinite (or even finite but has more elemnets than we want to list out), the use of ellipses is simply a convenience designed to help our intiutive understanding, but is not as mathematically precise as set-builder notation.

3. (5 pts) Define the set $\mathcal{N}^{1mod3}$ of all natural numbers that can be written as $3n+1$ for some natural number n. These are the numbers that produce a remainder 1 when divided by 3.

   Solution

   Using ellipses: $\mathcal{N}^{1mod3} = \{1, 4, 7, 10, 13, ...\}$

   Using the set builder notation: $\mathcal{N}^{1mod3} = \{k | k = 3xn + 1, \text{ for some } n \in \mathcal{N}\}$

4. A set S is said to be *closed* under an operation if the result of applying that operation to one or more elements of that set is always in the set. (How many elements the operation is applied to depends on how many operands that operation takes.)

   (a) (5 pts) Is $\mathcal{N}$ closed under addition? Is it closed under subtractions? Explain briefly (no rigorous proof is required).

   Solution

   It is closed under addition - as a sum of two natural numbers is again a natural number.

   It is not closed under subtractions. For example, $3 \in \mathcal{N}$, $5 \in \mathcal{N}$, but $3 - 5 = -2 \notin \mathcal{N}$.

   (b) Prove or disprove (rigorously):

   - (5 pts) $\mathcal{N}^{1mod3}$ is closed under addition.
     Solution
     $\mathcal{N}^{1mod3}$ **is not closed under addition**
     Proof by a counterexample
     Suppose $k_1 = 3xn_1 + 1$ and $k_2 = 3xn_2 + 1$ are two elements of the set $\mathcal{N}^{1mod3}$. Then the sum
     $k_1 + k_2 = 3xn_1 + 1 + 3xn_2 + 1 =$
     $3x(n_1 + n_2) + 2$
     which cannot be an element of $\mathcal{N}^{1mod3}$. *QED*
   - (5 pts) $\mathcal{N}^{1mod3}$ is closed under multiplication.
     Solution
     $\mathcal{N}^{1mod3}$ **is closed under multiplication**

Proof by construction

Suppose $k_1 = 3xn_1 + 1$ and $k_2 = 3xn_2 + 1$ are two elements of the set $\mathcal{N}^{1mod3}$. Then the product
$k_1xk_2 = (3xn_1 + 1)x(3xn_2 + 1) =$
$9n_1n_2 + 3n_1 + 3n_2 + 1 =$
$3x(3n_1n_2 + n_1 + n_2) + 1 =$
$3xn' + 1$ where $n' = 3n_1n_2 + n_1 + n_2 \in \mathcal{N}^{1mod3}$. *QED*

5. (5 pts) Define the set $\mathcal{N}^{2mod3}$ of all natural numbers that can be written as $3n + 2$ for some natural number n.

   Define the set $\mathcal{N}^{0mod3}$ of all natural numbers that can be written as $3n$ for some natural number n.

   Give a rigorous proof that the set $\mathcal{N}^{0mod3} \cap \mathcal{N}^{1mod3} \cap \mathcal{N}^{2mod3}$ is the empty set.

   Solution

   Proof by a counterexample

   Suppose $k \in \mathcal{N}$ is an element in the set $\mathcal{N}^{0mod3} \cap \mathcal{N}^{1mod3} \cap \mathcal{N}^{2mod3}$. Then there exist numbers n_0, n_1, n_2 such that

   $k = 3xn_0$

   $k = 3xn_1 + 1$

   $k = 3xn_2 + 2$

   That means

   $3xn_0 = 3xn_1 + 1$

   or

   $3x(n_0 - n_1) = 1$, which is false.

   Also,

   $3xn_0 = 3xn_2 + 2$

   or

   $3x(n_0 - n_2) = 2$, which is false.

6. (5 pts) Give a rigorous proof that the set $\mathcal{N}^{0mod3} \cup \mathcal{N}^{1mod3} \cup \mathcal{N}^{2mod3}$ is the set of all natural numbers $\mathcal{N}$.

   Solution

   Proof by Induction:

The base case:

We know $0 = 3x0 \in \mathcal{N}^{0mod3}$, and so

$0 = 3x0 \in \mathcal{N}^{0mod3} \cup \mathcal{N}^{1mod3} \cup \mathcal{N}^{2mod3}$,

The induction hypothesis:

We know that for some $n \in \mathcal{N}$ it is true that

$n \in \mathcal{N}^{0mod3} \cup \mathcal{N}^{1mod3} \cup \mathcal{N}^{2mod3}$.

Proof of the statement for $n + 1$:

We need to show that for $n + 1$ is also true that

$n + 1 \in \mathcal{N}^{0mod3} \cup \mathcal{N}^{1mod3} \cup \mathcal{N}^{2mod3}$.

If $n \in \mathcal{N}^{0mod3} \cup \mathcal{N}^{1mod3} \cup \mathcal{N}^{2mod3}$ then one of the following is true:

$n = 3xk$ for some $k \in \mathcal{N}$

$n = 3xk + 1$ for some $k \in \mathcal{N}$

$n = 3xk + 2$ for some $k \in \mathcal{N}$

But then one of the following holds:

$n + 1 = 3xk + 1$ for some $k \in \mathcal{N}$

$n + 1 = 3xk + 2$ for some $k \in \mathcal{N}$

$n + 1 = 3xk + 3 = 3x(k + 1)$ for some $k \in \mathcal{N}$

and so $n + 1 \in \mathcal{N}^{0mod3} \cup \mathcal{N}^{1mod3} \cup \mathcal{N}^{2mod3}$. QED

7. (10 pts) The Indigo Country has a number of train companies providing transportation within this large country. The minister of transportation proudly claims that at least one of the train companies provides a dining car in every express train that has at least four wagons for passengers.

   (a) A disgruntled citizen of Indigo Country asserts that this claim is not true because he knows that the ExTrack Company runs the Orient Express train with five passenger cars and no dining car. Does this logic refute the claim? Explain clearly why or why not.

   Solution

   This statement does not refute the claim.

   The claim is that there is at least one such company — not that all companies have this property, so finding one that does not have the property is not sufficient.

(b) Another disgruntled citizen asserts that this claim isn't true becaue she knows that every train company runs at least one train that is not an express train. Does this logic refute the claim? Explain clearly why or why not.

Solution

This statement does not refute the claim.

The statement that every company runs at least one local train does not say anything about the express trains these companies run.

(c) If neither of these arguments refutes the minister's claim, explain exactly what needs to be done to prove that the claim is false.

Solution

We first make a list $SExp$ that includes every company that runs express trains. We choose the subset $SExp4$ of those companies that run express trains with at least four passenger wagons.

For each company in the set $SExp4$ we make a list $LExp4Dining_{company}$ of all express trains with at least four passenger wagons that do not contain a dining car.

The claim is refuted if for every *company* the set $LExp4Dining_{company}$ is not empty (i.e. for every company we can find at least one express train with four or more passenger cars that does not have a dining car.

8. (10 pts) Spiffy Chips Company Inc., has invented a special purpose computer chip, the Lexian, designed with only a limited set of capabilities. There are some things it cannot do at all. For example, it is known that it is not possible to write a program for a Lexian to combandle an arbitrary polybradik.

   (a) Algorithm designer Alla Ghowarizmi is investigating programs that rebluzeb arbitrary quentiglubs. She has shown how to build such programs using a polybradik-combandling program as a subroutine. Does it follow that it is not possible to write a program for a Lexian that rebluzebs an arbitrary quentiglub? Explain clearly why or why not.

   Solution

   It still can be possible to write a program for a Lexian that rebluzebs an arbitrary quentiglub.

The fact that the program that rebluzebs an arbitrary quentiglub has been written using a polybradik-combandling program as a subroutine does not mean that this is the only way the program can be written. Onother solution may not use the polybradik-combandling program, and the different approach could run on a Lexian chip.

(b) Another algorithm developer, Davy Loper, is interested in programs to twiglimate arbitrary karedias. He has shown that, given any program that can do this, such a program can be called with certain particular arguments (on any processor) to combandle an arbitrary polybradik. What does this imply about the possibility of writing a program on a Lexian that twiglimates an arbitrary karedia. Prove your answer.

Solution

We cannot write a program for Lexian that twiglimates an arbitrary karedia.

If we could, then this program could be used to combandle an arbitrary polybradik (if given the appropriate arguments), but that would mean there is a program for Lexian that combandles an arbitrary polybradik, contrary to the limitations of the Lexian.

Hint: There are three main statements to consider the truth or falseness of and/or relations between:

- A Lexian can be programmed to combandle an arbitrary polybradik.
- A Lexian can be programmed to rebluzeb an arbitrary quentiglub.
- A Lexian can be programmed to twiglimate an arbitrary karedia.

9. (10 pts) Suppose in Java you have the following interface and class definitions: (assume the standard full constructors for all classes)

```
interface Comparable{
  int compareTo(Comparable that); }

class Book implements Comparable{ ...}

interface IShape{ }
class Circle implements IShape{
  Posn loc;
  int radius;
```

```
  ...
}
class Square implements IShape{
  Posn loc;
  int size;
  ...
}

interface LoS extends Comparable{ }
class MTLoS implements LoS{ }
class ConsLoS implements LoS{
  IShape first;
  LoS rest;
  ...
}
```

(a) Each class and interface definition defines a class of data objects. Write down a list of all subset relationships that are known from these definitions. For example, $Circle \subseteq IShape$.

Solution

$Book \subseteq Comparable$

$Circle \subseteq IShape$

$Square \subseteq IShape$

$LoS \subseteq Comparable$

$MTLoS \subseteq LoS$

$ConsLoS \subseteq LoS$

$MTLoS \subseteq Comparable$

$ConsLoS \subseteq Comparable$

(b) Suppose the programmer defines the following variables and follows with the shown assignments. For each assignment indicate whether it is a valid statement. If not, explain why the compiler will reject it.

```
// Definitions:
IShape c;
IShape s;
Circle cir;
Comparable b;
Comparable cmp1;
Comparable cmp2;
LoS lst1;
LoS lst2;
LoS lst3;
LoS lst4;
```

```
LoS lst5;
MTLoS mt;

// Assignments:
c = new Circle(...);
c = new Square(...);
s = new Circle(...);
s = new Square(...);
cir = new Circle(...);
cir = new Square(...);
b = new Book(...);
cmp1 = new Book(...);
cmp2 = new Circle(...);
ls1 = new MTLoS();
ls2 = new ConsLoS(cmp1, ls1);
ls3 = new ConsLoS(cir, ls1);
ls4 = new ConsLoS(b, ls1);
ls5 = new ConsLoS(s, ls1);
mt = new ConsLoS(s, ls1);
```

(c) If we eliminate all invalid assignments, show for each variable in a valid assignment the class that represents its current runtime type. (When the object represented by this variable invokes a method, the search for method definition begins in this class.)

Solution

```
// Assignments:
c = new Circle(...);             Circle
c = new Square(...);             Square
s = new Circle(...);             Circle
s = new Square(...);             Square
cir = new Circle(...);           Circle
cir = new Square(...);           illegal statement
b = new Book(...);               Book
cmp1 = new Book(...);            Book
cmp2 = new Circle(...);          illegal statement
ls1 = new MTLoS();               MTLoS
ls2 = new ConsLoS(cmp1, ls1);    illegal statement
ls3 = new ConsLoS(cir, ls1);     ConsLoS
ls4 = new ConsLoS(b, ls1);       illegal statement
ls5 = new ConsLoS(s, ls1);       ConsLoS
mt = new ConsLoS(s, ls1);        illegal statement
```

Additional formatting samples

1. Give the state transition diagram for the FA whose formal description is $(\{1, 2, 3, 4\}, \{a, b\}, \delta, 1, \{1, 4\})$, where δ is given by the following

table:

	a	b
1	1	2
2	3	4
3	2	1
4	2	4

2. Give a formal description of the FA1 shown in Figuree 1:

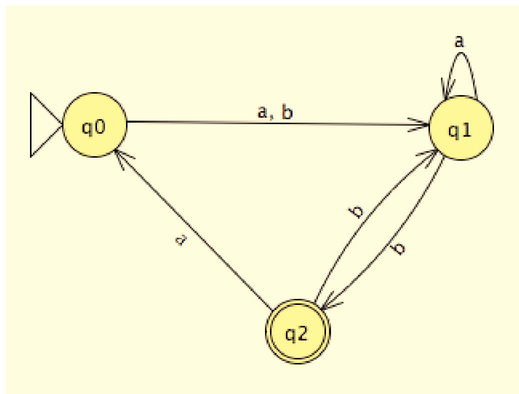

Figure 1: FA1