Math 55 - Spring 2004 - Lecture notes # 3 - Jan 27 (Tuesday)

Keep Reading: Sections 1.6, 1.7 and 1.8
Homework: section 1.6: 8, 12, 16, 22
section 1.7: 4, 26, 38, 40 for sets {2,4,6,8} and {1,3,5,7}
section 1.8: 6, 16, 22, 26, 32, 36, 64

Today's goals: sets
functions
countability
can one infinite set be bigger than another?
are there programs to compute all possible functions?

But first, a few words about new reasons proofs matter:

It is possible to embed proofs in downloadable programs and email, so that the recipient is guaranteed (by running a program to check the proof) that the program or email does not contain a virus, or will not otherwise cause mischief.

For example, Java enabled cell-phones in Japan use this technique of "proof-carrying code" when they download code. When a phone downloads Java code it checks the proof that the code will not cause damage (eg write into memory it should not). The alternative is to run them using an interpreter that keeps downloaded code from causing mischief. But it uses fewer instructions (and so less battery power!) to check a proof and then run noninterpreted code. Prof. George Necula has pioneered this idea (www.cs.berkeley.edu/~necula)

As another example, last week Bill Gates proposed a list of techniques to combat spam, including proof-carrying-email, which would contain a proof that the email did not come from a spammer, i.e. sent out to lots and lots of people. One way to do this would be for the email to contain a very difficult puzzle and its solution, where the puzzle is known to be so difficult that only by spending at least, say, 1 CPU second could the puzzle have been solved, although it is easy for your mailer to check that the solution is correct. Thus, sending a billion spam messages would cost a billion CPU seconds instead of a few seconds, making span uneconomical to send.

First goal: sets

```
DEF: a set A is a collection of elements, also called members
EG: A = {a,b,c}, A = {1,2,3,...,100},
    A = { k^2 | k an integer, 1 <= k <= 100}
      = "the set of numbers k^2 such that k is an integer between 1 and 100"
EG: Z = integers, N = nonnegative integers, Q = rational numbers,
    R = real numbers, R+ = nonnegative reals
DEF: A = B as sets if same elements (order doesn't matter, multiple
     copies of elements does not matter, so {1,1,2} = {1,2} = {2,1})
DEF: NULLSET = {} = set with no elements
DEF: x IN A is a proposition which is true if x is an element of A
DEF: A SUBSET B means (x IN A) -> (x IN B)  (Venn diagram)
DEF: Intersection: "C = A inter B" means
        forall x, (x IN C) <-> (x IN A) and (x IN B)
     "C = A1 inter A2 inter ... An = inter_{i=1}^n Ai" means
        forall x, (x IN C) <-> ( (x IN A1) and (x IN A2) and ... and
                                 (x IN An) )
     (Venn diagram)
DEF: A inter B = NULLSET mean A and B are disjoint
DEF: Union: "C = A union B" means
        forall x, (x IN C) <-> (x IN A) or (x IN B)
     "C = A1 union A2 union ... An = union_{i=1}^n Ai" means
        forall x, (x IN C) <-> ( (x IN A1) or (x IN A2) or ... or
                                 (x IN An) )
     (Venn diagram)
DEF: The Universal Set is the set of all possible elements a set can have
EG:  If U = "integers", A SUBSET U can be odd integers, perfect squares, etc.
DEF: The complement of A, written bar(A) (A with a bar over it),
     is the set of all elements of U not in A
Theorem (DeMorgan):  bar(A union B) = bar(A) inter bar(B)
     Proof: (1) Venn Diagram
            (2) show that propositions x IN bar(A union B) and
                x IN bar(A) inter bar(B) are logically equivalent,
                using DeMorgan's law:
                x in bar(A union B) <=>
                not(x in A union B) <=>
                not (x in A or x in B) <=> (use DeMorgan for propositions)
                not(x in A) and not(x in B) <=>
                x in bar(A) and x in bar(B) <=>
                x in bar(A) inter bar(B)
DEF: |A| = cardinality of A = number of elements in A
     (if A is finite, else say A is infinite,
      details of infinite case later)
```

```
   DEF: P(A) = power set of A = set of all subsets of A
   EG: A={1,2}, P(A)={NULLSET, {1}, {2}, {1,2}}
ASK&WAIT: if |A| is finite, what is |P(A)|?
          Why: Write A = {a1,a2,...,an} where n=|A|,
               Can uniquely identify any B SUBSET A by n bits b1,...,bn
               where bi=1 if ai IN B, and bi=0 if ai NOT_IN B
               There are clearly 2^n such bit strings
   DEF: (a1,...,an) is an ordered n-tuple
         ( parentheses() instead of braces{} means order matters)
        (a1,a2) is also called an ordered pair
   DEF: If A and B are sets, then A x B = { (a,b) | a in A and b in B }
        is the Cartesion product of A and B. A x B is also called the
        set of ordered pairs (a,b) from A, B
   EG: Suppose A and B are both set of real numbers.
       Then A x B is a 2-dimensional plane
   DEF: If A1, A2, ... , An are sets then
        A1 x A2 x ... x An = { (a1,a2,...,an) | ai in Ai for i=1,..,n }
        is the Cartesion product of A1,...,An. It is also called the
        set of ordered n-tuples (a1,...,an) from A1,..,An
   EG: A = {all keys on a keyboard, including return}
         = {a,b,...,z,A,B,...,Z,0,...,9,@,#,..,}
       A^2 = A x A = all ordered pairs of characters
       A^n = A x A x ... x A (n times)
           = all ordered n-tuples of characters
       S = A U A^2 U A^3 U ...
         = all finite strings of characters
       E = all syntactically correct English sentences
       E subset S
       J = all syntactically correct Java programs
       J subset S

   Second goal: functions

   DEF: Let A and B be sets. A function f from A to B (write f:A->B) is
        an assignment of exactly one element of B to each element of A
        (write f(a)=b to mean b IN B is assigned to a IN A). A is called
        the domain of A, and B is called the codomain.
        b=f(a) is the image of a, a is the preimage of b
        { f(a) | a IN A } is called the range of f
        (Figures 1, 2 in sec 1.8 show how functions may be represented)
   EG:  f:Z->Z where f(z) = z^2,
   EG:  f:Z->Z, where f(z) = 1, "constant function"
```

```
   EG:  FLOOR:R->Z, where FLOOR(x) = largest integer <= x
   EG:  CEILING:R->Z, where CEILING(x) = smallest integer >= x
   EG:  LOG_2:R+->R, where R+ = nonnegative reals,
        LOG_2 = logarithm base 2 of x
   EG:  f:{workers}->Z, where f(worker) = worker's Social Security # (SS#)
        (represented by table, not formula)
   EG:  f:Z->{integer multiples of .01} where f(account number) = balance
ASK&WAIT function f1(x), return x, end,
         function f2(x), return (2*x)/2, end
         What are domain and codomain?
         How can we choose the domain and codomain to make these functions equal?
         Are they the same functions on a computer?
   EG:  Suppose f1:A->R and f2:A->R where A is any set, then
        f=f1+f2, g=f1*f2 etc are the functions satisfying
        f(x)=f1(x)+f2(x),g(x)=f1(x)*f2(x), etc
ASK&WAIT: Let f1:R->R, where f1(x)=x, f2=f1, and f3:R->R, f3(x)=1.
          Does f1/f2 = f3 as functions?
          Generally, when is h=f1/f2 a function?
          How can we change h slightly to make it a function?
          What happens if we implement f1(x)/f2(x) on a computer?
   EG:  B = {functions f_z:Z->Z | f_z(x)=x+z, z IN Z }, a set of functions
        g:Z->B, g(z) = f_z, i.e. function that adds z to its argument:
        g(z)(x) = f_z(x) = z+x

   DEF: f:A->B is one-to-one (injective) if f(x)=f(y) -> x=y
ASK&WAIT  is f:N->N where f(x) = x^2, injective?
ASK&WAIT  is f:Z->Z where f(x) = x^2, injective?
ASK&WAIT  is f:{workers}->Z where f(x) = SS#, injective?,

   DEF: f:A->B is onto (surjective) if all b IN B have preimages in A
ASK&WAIT  is f:Z->Z where f(x)=x+1 surjective?
ASK&WAIT  is f:N->N where f(x)=x+1 surjective?
ASK&WAIT  is f:{workers}->{9 decimal digit integers}, f(x) = SS#,
          surjective?
ASK&WAIT: What does it mean if
          f:{drivers license numbers}->{names of actual drivers}
          where f(license number) = name of driver on license
          is not surjective?

   DEF: f:A->B is a one-to-one correspondence (bijective)
        if it is one-to-one and onto
ASK&WAIT: is f:Z->Z where f(x)=x+1 bijective?
```

```
ASK&WAIT: is f:Z->{even integers} where f(x)=2*x bijective?

   DEF: If f:A->B is a bijection, then the function f^{-1}:B->A defined
        by f{-1}(b)=a if f(a)=b is called the inverse function of f
ASK&WAIT: if f:Z->Z, f(x)=x+1, what is f^{-1}?
ASK&WAIT: if f:Z->{even integer}, f(x)=2*x, what is f^{-1}?
ASK&WAIT: if f:{drivers license numbers}->{names of drivers},
          what is f^{-1}?

   DEF: If g:A->B and f:B->C, then the function h:A->C defined by
        h(a)=f(g(a)) is called the composition of f and g, written h=f o g
ASK&WAIT: f:R->R+, g:R+->R, (R+ = nonnegative reals)
          f(x)=x^2, g(x) = sqrt(x)
          What is f o g? (domain, codomain, value)?
          What is g o f?
   DEF: id_A:A->A is the "identity function", if id_A(a)=a for all a IN A
ASK&WAIT: let f:A->B be a bijection, and f^{-1}:B->A be inverse of f.
          What is f o f^{-1} ?
          What is f^{-1} o f?
   EG:  f:{1,2,...,26}->{a,b,...,z} with f(1)=a,...,f(26)=z
        f o f^{-1} is identity on {a,b,...,z}
        f^{-1} o f is identity on {1,2,...,26}
   DEF: If f:A->B, then the graph of f is the set of ordered pairs
        { (a,b) | a IN A and f(a)=b }
ASK&WAIT: what is graph of f:R->R, f(x)=x^2?
ASK&WAIT: what is graph of f:{workers}->Z, f(worker)=SSN?


   Third Goal: understand cardinality, countability

   Recall DEF: If A is finite, the cardinality |A| = # members of A
ASK&WAIT: suppose f:A->B is a bijection, A finite. Is B finite?
          How are |A| and |B| related?
   DEF: We say that A and B have the same cardinality if there is a
        one-to-one correpondence between them, whether finite or not
ASK&WAIT: Do Z and {even integers} have same cardinality?
ASK&WAIT: Do N and {powers of 2} have same cardinality?
ASK&WAIT: Do Z and N have same cardinality? (hint: represent bijection by table)
   DEF: A set that is either finite or has the same cardinality
        as N (or Z) is called countable, else uncountable
        intuition is that an uncountable set is much larger than
         any countable set
```

More examples of countable sets (most sets we have seen are countable:)

Theorem: if A and B are countable, so is S = A union B
proof: number elements of S by a(1), b(1), a(2), b(2),...
i.e. f(i) = a(i/2) if i is even; b((i+1)/2) if i is odd
is a one-to-one correspondence between N and S
Enough to illustrate bijection f:N->S of a set S with N or Z without writing down formula for f, i.e. just show how to write down all members of S in order each member of S appearing exactly once

Theorem: The Cartesian product P = A x B of all pairs {(a,b)}
is countable if A and B are countable
proof: represent P as "lattice" points in the plane,
and number them diagonally.

Theorem: Suppose A1, A2, A3, ... are all infinite countable sets
Then S = A1 union A2 union A3 union ... is countable
ASK&WAIT: why?

Theorem: Suppose A is countable, and B is a subset of A.
Then B is countable
ASK&WAIT: why?

ASK&WAIT: Is Q (rational numbers) countable?

Theorem: Suppose A1, A2, A3, ... are all countable sets
Then S = A1 union A2 union A3 union ... is countable
ASK&WAIT: why?

ASK&WAIT: Is J (set of syntactically correct programs) countable?