0. Real Analysis, the Last Crusade

I think one of the biggest illusions the brain gives us is that we remember everything as we read. But our memory is more like a FIFO that keeps only the last 7 things in RAM, discarding everything before. that is why re-reading passages, even re-reading the beginning of the paragraph, is essential to remember what is going on. Because I always forget what I'm even doing and need to remind myself what the current discussion is.

I am returning to Yeh's book after an infinite sequence of disappointments. There are exactly 0 books on Probability worth studying because they are either too hard, or have incomplete solutions. My plan here is to work through a book completely, do all the exercises, and check them as I go. Just like I did with Abbott. Remember this is a graduate book and is not meant to be easy. But the hard part is going to be the math itself, not the incompetency of the author.

I am really in a corner here. This is the most core subject needed to be a graduate student. It is all or nothing. Either I succeed at this and become a mathematician, or I become a soy web developer. There will be no more book switching because I literally tried 4 other books (Capinski, Rosenthal, some 2020 book, Resnick), hoping they would be easier, and they all suck ass. Schilling is great as an appetizer. But it is time to get serious.

Additionally I was confused, but this seems to be the same content as Rudin, Royden, Folland, and such. Meaning this book is what first year graduate mathematicians are taking in class. If I finish this, I will be good to study all sorts of goodies like functional analysis, probability, etc.

Math books are either boring or confusing. They are boring if everything is laid out to you, and as you read, you think "hmm.. yes, yep, i see how this follows from that. yawn". A confusing book is one where the author leaves out sections of reasoning because they are obvious to them. When reading, you would think: "...wait, what? Am I missing something? how did they go from line n to line n+1? Am I retarded? Should I give up mathematics and work at McDonalds?". A boring book is called boring because it lacks this emotional roller coaster, as well as other adventures like looking for tutors, cheating, googling ways to end your life, etc..

1.0 Measure Spaces

```
consider measuring a subset of R.
    for an interval (a,b), length is b-a
    for infinite interval, length is infinity.

    for any set E in Power(R),
        let I_n be seq of open intervals covering E.
        take the sum of these intervals' lengths.
```

```
        define mu* as the infimum of all such sums

    0 <= mu* <= inf
        mu*(nullset) = 0
        monotone: mu*E <= mu*F if E subset F
        mu*I = length(I) for interval I
        subadditivity: mu*(E1 U E2) <= mu*E1 + mu*E2

    additivity: v(E1 U E2) = v(E1) + v(E2)   for disjoint
        requires E1,E2,(E1 U E2) in the collection

    mu* is not additive on Power(R), there are wack sets
        (sets that are disjoint but not separate enough)

    it is additive if we only consider good sets
    E in Power(R)
    for A in Power(R),
        (A int E) U (A int Ec) = A  are disjoint

E is mu* measurable if
    mu*(A) = mu*(A int E) + mu*(A int Ec)
        forall A in Power(R)
        (so E,Ec partition A into additive slices.
        we don't have the problem that E,Ec are
        disjoint but not separate enough)
    nullset, R satisfy this.

MM id collection of all measurable sets.
    closed under union:
        E1,E2 in MM
        mu*(A) = mu*(A int E1) + mu*(A int E1c)
        mu*(A) = mu*(A int E2) + mu*(A int E2c)
            rewrite A as A int E1c
        mu*(A E1c) = mu*(A E1c E2) + mu*(A E1c E2c)
            put this thing into E1's condition
        mu*(A) = mu*(A E1) + mu*(A E1c E2) + mu*(A E1c E2c)

        (A E1) U (A E1c E2)
            A (E1 U (E1c E2))
            A (E1 U (E2 \ E1))
            A (E1 U E2)
            now we know
            A (E1 U E2) = (A E1) U (A E1c E2)

        by subadditivity,
```

```
        (remember, we don't have additivity,
            only measurability)
        mu*(A (E1 U E2)) <= mu*(A E1) + mu*(A E1c E2)
            and similarly,
        mu*(A) >= mu*(A (E1 U E2)) + mu*(A E1c E2c)
            note (E1c E2c) = (E1 U E2)c
        mu*(A) >= mu*(A (E1 U E2)) + mu*(A (E1 U E2)c)
            reverse holds by subadditivity.
        mu*(A) <= mu*(A (E1 U E2)) + mu*(A (E1 U E2)c)
            thus we get
        mu*(A) = mu*(A (E1 U E2)) + mu*(A (E1 U E2)c)
            so (E1 U E2) is measurable.
    thus MM closed under union.

collection of subsets of nonempty set X is called a
    sigma-algebra of subsets of X, if:
    - X in it
    - complements
    - countable union

    MM is sigma-algebra of subsets of R.
        (basically "of subsets of _" means _ is the main set)

mu* additive on MM
    E1, E2 in MM disjoint
        put (E1 U E2) in measurability condition of E1:
    mu*(E1 U E2) = mu*((E1 U E2) E1) + mu*((E1 U E2) E1c)
        note
        ((E1 U E2) E1) = E1   (disjoint)
        note
        ((E1 U E2) E1c) = E2
        thus
    mu*(E1 U E2) = mu*(E1) + mu*(E2)
```

1.1 Measure on a sigma algebra of sets

```
ALGEBRA: X is a set. collection AA of subsets of X is
    an algebra / field of subsets of X if:
    - X in AA
    - complements
    - finite union

    properties:
    1) nullset in AA
    2) union 1..n
```

3) intersection
4) int 1..n
5) A \ B in it, if A,B in it

proof:
1) X in AA, and Xc = nullset
2) keep repeating A U B finite number of times
3) A int B = (Ac U Bc)c
4) repeat intersection finite number of times
5) A \ B = A int Bc

SIGMA-ALGEBRA: collection of subsets of X, is an algebra, also
- countable union

if an algebra is a finite collection, it is a sigma algebra
(since there is no countable seq of sets to unify)
(this assumes the entire algebra is finite, after all
possible finite union and complement operations)

- countable intersection
pf:
int An = (U Anc)c

A in Power(X) is same as A subset X

Power(X) is the largest sigma-algebra, meaning that if
AA is another sigma-algebra of subsets of X,
and if Power(X) subset AA, then AA = Power(X)

{nullset, X} is smallest sigma-algebra, meaning if
AA is sig-alg of X and AA subset {nullset, X}, then
AA = {nullset, X}

in R^2, RR is rectanble (a1,b1]x(a2,b2]
with -inf <= ai < bi <= inf
let AA be collection of finite unions
RR subset AA
since each rect is union of 1 rect
say nullset is union of 0 rectangles
nullset in AA
AA is algebra of subsets of R^2
but its not a sigma-alg
consider infinite checkerboard tiling, it is not in AA.

Limits of Sequences of Sets

```
An sequence of subsets.
    increasing: An ^ if An  subset  An+1
    decreasing: An v if An+1  subset  An
    monotone: it is eithe increasing or decreasing

    increasing sequence:
        lim An := U_n An = {x : x in An for some n}
            example: [0, n/n+1] -> [0,1)]

    decreasing sequence:
        lim An := int_n An = {x : x in An for each n}
            example: [0,1+e) -> [0,1]
            example: (0,1/n) -> nullset
            example: [0,1/n) -> {0}

    limit always exists for monotone sequence.

    liminf An := U_n int_k>n Ak
        note: int_k>n Ak is increasing seq, thus:
        lim int_k>n Ak = U_n int_k>n Ak
        exists because we defined limits of inc seq

    limsup An := int_n U_k>n Ak
        note: U_k>n Ak is decreasing, thus:
        lim U_k>n Ak = int_n U_k>n Ak
        exists because we defined limits of dec seq

        [i just realized. the outer op is equivalent to
        limit, since the inner set is inc/dec. but the
        inner op set is actually analogous to the
        min/max of a number sequence. it's the same
        as analyzing an oscillating sequence, we need
        to check that the limit of the min and limit
        of the max are the same, otherwise we get
        something like +1,-1,+1,-1,...
        and in this case the intersection represents
        the smallest set while union represents the
        largest set. they both need to converge to
        the same thing for the limit to exist.]

An sequence of subsets.
    1) liminf An = {x : x in An for all but finitely many n}
    2) limsup An = {x : x in An for infinitely many n}
```

```
3) liminf An  subset  limsup An

pf:
1)
    <-
    if x in An for all but finitely many n,
    then there exists n0 st x in Ak forall k>n0.
    [since it's a finite number, just pick the last one]
    thus x in int_k>n0 Ak
    thus x in int_k>n0 Ak  subset  U_n int_k>n Ak
    thus x in liminf An
    ->
    if x in liminf An, x in int_k>n Ak for some n,
    thus x in Ak forall k>n. that means x in An forall
    but finitely many n.
2)
    <-
    if x in An for infinitely many n, then for each n
    x in U_k>n Ak
    [think of it as we enumerate all occurences of x.
    so the first one is at position >= 1, second is at
    >= 2, i^th one is >= i.]
    thus x in int_n U_k>n Ak = limsup An
    ->
    if x in limsup An, then x in U_k>n Ak for each n.
    thus for every n, x in Ak for some k>n.
    thus for every n, x in some further Ak.
    thus x in An for infinitely many n
3)
    liminf implies x occurs all but finitely many times.
    that means it occurs an infintie number of times.
    that means it is a subset of limsup.

for arbitrary set sequence An, converges if limsup = liminf
    lim An := liminf An = limsup An
    if liminf =/= limsup, limit does not exist

    this def collapses to the monotone version if An is monotone:
    if An ^,
        int_k>n Ak = An
        U_n int_k>n Ak = U_n An
        liminf = U_n An

        U_k>n Ak = U_k Ak (union starting at n = union over all)
        int_n U_k>n Ak = U_k Ak
```

```
        limsup = U_n An

        thus limsup = liminf = U_n An = prev def
    if An v,
        int_k>n Ak = int_k Ak (int from n = int over all)
        U_n int k>n Ak = U_n int_k Ak = int_k Ak
        liminf An = int_n An

        U_k>n = An
        int_n U_k>n Ak = int_n An
        limsup An = int_n An

        thus limsup = liminf = int_n An = prev def

example:
    i odd: An = [0, 1/i]
    i even: An = [0,i]
    liminf = {0}
    limsup = [0,inf)
        [note: here we have union [0,i] forall i even.
        it includes all integers, but not infinity.
        since infinity itself is never achieved at some i.
        so the interval [0,inf) basically represents
        that exact union of all finite intervals, but
        never quite reaching infinity]

    [i usually imagine the intersection of the tail
    and the union of the tail as n->infinity]

Thm 1.9
    AA sigma-algebra
    for any sequence An in AA, liminf and limsup in AA
        (and so is limit, if limsup = liminf)
    pf:
        know An in AA.
        (int_k>n Ak) in AA, by countable int, for each n
        U_n (int_k>n Ak) in AA, by countable union.
        thus liminf in AA.

        (U_k>n Ak) in AA by countable union, for each n
        int_n (U_k>n Ak) in AA by countable int
        thus limsup in AA
```

Generation of Sigma-Algebras

```
Lem 1.10
    {AA_a : a in A} collection of sigma-algebras of subsets of X
    then int_a AA_a is a sigma-algebra of subsets of X.
    same result for algebras.

    pf:
    1) X in AA_a for every a, so X in intersection.
    2) if E in intersection, then E in AA_a for each a,
        and for each a, AA_a contains Ec.
        thus Ec in intersection.
    3) if En in intersection, En in AA_a for each a.
        and for each a, AA_a contains U En.
        thus U En in intersection.

Thm 1.11
    CC collection of subsets of X.
    there exists the smallest sigma-algebra AA0 containing CC.
        meaning: if AA is a sigma-algebra containing CC,
        then AA0  subset  AA
    same for algebras.

    pf:
    Power(X) is one such sigma-algebra, so the set is non-empty.
    let {AA_a : a in A} be
        collection of all sigma-algebras of X containing CC.
    then int_a AA_a contains CC and is sigma-algebra.
    it is the smallest sigma-algebra containing CC,
        because any other sigma-algebra containing CC would
        be in {AA_a : a in A}
        and int_a AA_a  subset  AA_a  forall a

for arbitrary collection CC of subsets of X,
    sigma(CC) is the smallest sigma-algebra of X containing CC.
    called "sigma-algebra generated by CC"
    similarly alpha(CC) is algebra generated by CC.

if CC1, CC2 are collections of subsets of X,
    and CC1  subset  CC2,
    then sigma(CC1)  subset  sigma(CC2)
        Quik Proof:
        CC1  subset  CC2  subset  sigma(CC2)
        sigma(CC1) = int_a {AA_a : AA_a contains CC1}
        so sigma(CC2) is included in the RHS set
```

thus sigma(CC1) subset sigma(CC2)]

if AA is a sigma-algebra of X,
sigma(AA) = AA
Quik Proof:
int_a {AA_a : AA_a contains AA} = AA
<-
AA contained in each AA_a, so in intersection too.
->
know AA is in this set bc it is sigma-algebra.
if x in LHS, x in subset AA subset RHS

in particular,
sigma(sigma(CC)) = sigma(CC)

f maps X to Y.
image is f(X) subset Y
E subset Y
E need not be subset of f(X)
can be disjoint from f(X)
pre-image of E under f is subset of X:
f^-1(E) := {x in X : f(x) in E}
if E int f(X) = nullset,
then preimage is nullset.
[f does not map any x into E]
for arbitrary E subset Y,
f(f^-1(E)) subset E
[shouldn't it be equal? we are considering
all x that get mapped to E, so f(x) will
cover all of E. unless E is like a "range"
meaning that only part of E is mapped into.]

f^-1(Y) = X
[from the entire range, we get the domain]

f^-1(Ec) =
f^-1(Y \ E)
f^-1(Y) \ f^-1(E)
[schilling: f^-1 works across all set ops]
X \ f^-1(E)
(f^-1(E))c

f^-1(U Ea) = U f^-1(Ea)

f^-1(int Ea) = int f^-1(Ea)

```
    for collection CC of subsets of Y,
    f^-1(CC) := {f^-1(E) : E in CC}

Prop 1.13
    f: X -> Y
    if BB is sigma-algebra of subsets of Y, then
        f^-1(BB) is a sigma-algebra of subsets of X.

    pf:
    1)
        Y in BB.
        X = f^-1(Y)
        thus X in f^-1(BB)
    2)
        A in f^-1(BB)
            thus A = f^-1(B) for some B in BB
        Bc in BB
        f^-1(Bc) in f^-1(BB)
        f^-1(Bc) = (f^-1(B))c = Ac
        thus Ac in f^-1(BB)
    3)
        An in f^-1(BB)
            thus An = f^-1(Bn) for osme Bn in BB
        U An = U f^-1(Bn) = f^-1(U Bn) in f^-1(BB)
            since U Bn in BB.

Them 1.14
    f: X -> Y
    arbitrary collection CC of subsets of Y
    sigma(f^-1(CC)) = f^-1(sigma(CC))
    pf:
    ->
    CC  subset  sigma(CC)
    f^-1(CC)  subset  f^-1(sigma(CC))
        quik proof:
        A  subset  B
            -> f^-1(A)  subset  f^-1(B)
        x in f^-1(A)
        f(x) in A
        f(x) in B since A subset B
        x in f^-1(B)
    sigma(f^-1(CC))  subset  sigma(f^-1(sigma(CC)))
        sigma(CC) is sigma-algebra of Y
        f^-1(sigma(CC)) is sigma-algebra of X
```

```
        so taking sigma operation doesn't change it
    sigma(f^-1(CC))  subset  f^-1(sigma(CC))
    <-
    let AA1 be arbitrary sigma-algebra of X
    let AA2 = {A subset Y : f^-1(A) in AA1}
        [sets whose preimages are in AA1]
        AA2 is sigma-algebra of Y
        1)
            X in AA1
            Y subset Y, satisfying f^-1(Y) = X
            thus Y in AA2
        2)
            A in AA2
            f^-1(A) in AA1
            (f^-1(A))c in AA1
            f^-1(Ac) in AA1
            Ac in AA2
        3)
            An in AA2
            f^-1(An) in AA1
            U f^-1(An) in AA1
            f^-1(U An) in AA1
            U An in AA2

        [TODO: is this the same as the set
        {f(A) : A in AA1}
        just try prove it i guess]

    in particular, let
    AA = {A subset Y : f^-1(A) in sigma(f^-1(CC))}
        [sets whose preimage in sigma(f^-1(CC))]
    CC  subset  AA
        [since preimage of CC will be in the
        sigma-algebra generated by preimages of CC]
    sigma(CC)  subset  AA
    f^-1(sigma(CC))  subset  f^-1(AA)
    f^-1(AA)  subset  sigma(f^-1(CC))  [by def of AA]
    thus f^-1(sigma(CC))  subset  sigma(f^-1(CC))

    [once again, we used a set defined by the property
    we want: that the preimage is in sigma(f^-1(CC)).
    why couldn't we work with that set directly?]

collection CC of subsets of X, arbitrary subset A of X:
    CC int A = {E int A : E in CC}
```

```
    write:
        sigma_A(CC int A)
        for sigma-algebra of subsets of A
        generated by collection (CC int A) of subsets of A
        subscript indicated the main set is A

Thm 1.15
    CC arbitrary collection of X
    A subset X
        sigma_A(CC int A) = sigma(CC) int A
    pf:

    ->
    CC  subset  sigma(CC)
    CC int A  subset  sigma(CC) int A
        apparently sigma(CC) int A
        is a sigma-algebra of A
        quik proof:
        1)
            A in sigma(CC)
            A int A = A
            thus A in it
        2)
            B in sigma(CC) int A
            X \ B in sigma(CC)
            (X \ B) int A in sigma(CC) int A
            ((X int A) \ (B int A))
                X int A = A
                B int A = B since B in sigma(CC) int A
            A \ B in sigma(CC) int A
        3)
            Bn in sigma(CC) int A
            Bn in sigma(CC)
            U Bn in sigma(CC)
            (U Bn) int A in sigma(CC) int A
            U (Bn int A) in sigma(CC) int A
                Bn int A = Bn since Bn in sigma(CC) int A
            U Bn in sigma(CC) int A
        [lol this was proved in Resnick but he just
        claimed this to follow]
    sigma(CC int A)  subset  sigma(CC) int A

    <-
    WTS: sigma(CC) int A  subset  sigma(CC int A)
```

```
let KK be collection of subsets K of X of type:
    K = (C int Ac) U B
        where
        C in sigma(CC)
        B in sigma_A(CC int A)
    note B  subset  A
    so B disjoint from Ac
    so union in K is disjoint
X in KK
    since let C=X, B=A,
    K = (X int Ac) U A = (X U A) int X = X
KK closed under countable unions
    Kn in KK
    Kn = (Cn int Ac) U Bn
    Un Kn = Un ((Cn int Ac) U Bn)
        = (Un (Cn int Ac)) U (Un Bn)
        = ((Un Cn) int Ac) U (Un Bn)
        know (Un Cn) in sigma(CC)
        know (Un Bn) in sigma_A(CC int A)
        so this thing in KK
KK closed under complements:
    Kc = X \ K
    [(X int Ac) U A] \ [(C int Ac) U B]
        (X int Ac), A disjoint
    (X int Ac) \ [(C int Ac) U B]
        U
        A \ [(C int Ac) U B]
    (X int Ac) \ [(C int Ac) U B]
        U
        A \ B
            since (C int Ac) disjoint from A
    (X int Ac) \ (C int Ac)
        U (A \ B)
            since B disjoint from (X int Ac),
            since B subset A
    [(X int Ac) \ (C int Ac)] U (A \ B)

    but
    (X int Ac) \ (C int Ac)
    (X int Ac) int (C int Ac)c
    (X int Ac) int (Cc U A)
    Ac int (Cc U A)
    ((Cc int Ac) U (A int Ac))
    (Cc int Ac)
```

```
    thus
    Kc = (Cc int Ac) U (A \ B) in KK

KK is sigma-algebra of X
note
    K int A = B in sigma_A(CC int A)
thus
    KK int A  subset  sigma_A(CC int A)

WTS: sigma(CC) int A  subset  KK int A
OR:  CC  subset  KK
    [this does not have to hold in general,
    perhaps it's only a subset WHEN intersected with A.
    but here he shows CC subset KK,
    which implies sigma(CC)  subset  KK,
    which implies sigma(CC) int A  subset  KK int A]

let E in CC
    E = (E int Ac) U (E int A)
    (E int A) in sigma_A(CC int A)
    E is of the form K
    E in KK
    CC subset KK

[i like the proof in Resnick page 19 much more.
it is fairly straightforward. It again uses the
technique where we define a set of subsets, this
time satisfying "set int A" in sigma(CC int A)]
```

Borel sigma-algebras

```
collection OO of subsets of X is a TOPOLOGY on X:
    - nullset in OO
    - X in OO
    - {E_a : a in A}  subset  OO  =>  U_a Ea in OO
        [arbitrary union]
    - E1, E2 in OO  =>  E1 int E2 in OO

    pair (X, OO) is a topological space.

    members of OO are called open sets.

    subset E of X is CLOSED if Ec is OPEN.
        X and nullset are both open & closed.
```

```
    arbitrary union of open sets is open.
    finite int of open sets is open.
    arbitrary int of closed sets is closed
    finite union of closed sets is closed

    interior oE of subset E is union of all open
        sets in E. it's the greatest open set in E.

    closure |E of E is intersection of all closed sets
        containing E. it's the smallest closed set containing E

    boundary of E = (oE U o(Ec))c
        [interior of E, interior of E's complement,
        then take the complement of those interiors]

    E is compact if every cover has a finite subcover
        cover is collection of open sets OO st
            E  subset  U V  for V in OO
        then there exists finite subcollection st
            E  subset U_n=1^N Vn

X any set. function p on XxX is a metric on X if:
    - p(x,y) in [0,inf)
    - p(x,y) = 0 iff x = y
    - p(x,y) = p(y,x)
    - triangle: p(x,y) <= p(x,z) + p(z,y)

    (X,p) is a metric space

    Euclidean distance is a metric.

    open ball:
        B(x0,r) = {x : p(x0,x) < r}
        E is called an open set if for each x in E,
        there is an open ball at x.

    collection of all open sets in a metric space satisfies
        the topology axioms. it's called the
        "METRIC TOPOLOGY of X by the metric p."

    E is bounded if there is a ball st
        E  subset  Ball

    E is compact iff it's closed and bounded.
```

OO collection of all open sets of X.
sigma(OO) is Borel sigma-algebra of subsets of X
its members are the Borel sets.

Lem 1.17
CC collection of all closed sets in topological space (X, OO)
then sigma(CC) = sigma(OO)
pf:
E in CC.
Ec in OO
Ecc in sigma(OO)
E in sigma(OO)
CC subset sigma(OO)
sigma(CC) subset sigma(OO)

E in OO
Ec in CC
Ecc in sigma(CC)
E in sigma(CC)
OO subet sigma(CC)
sigma(OO) subset sigma(CC)

E is Gdelta - int   of countably many open   sets.
E is Fsigma - union of countably many closed sets

if E is Gdelta, Ec is Fsigma.

they are Borel.

if E is Gdelta set,
E = int On
let Gn = int_k=1^n Ok
[intersection of the first n open sets]
Gn is decreasing sequence of open sets
int Gn = int On = E
thus a Gdelta set is always the limit of a decreasing
sequence of open sets.

[i suppose we don't consider the tail here like in
limsup, since we don't know if the tail will be
open or closed]

similarly, if E is Fsigma,
E = U Cn
where Cn closed set.

let Fn = U_k=1^n Ck
increasing sequence of closed sets
U Fn = U Cn = E
thus Fsigma is limit of increasing sequence of closed sets.

Measure on a sigma-algebra

CC collection of subsets of X.
y: CC -> [0,inf]
- monotone:
  y(E1) <= y(E2)   E1 subset E2
  E1, E2 in CC
- additive:
  y(E1 U E2) = y(E1) + y(E2)   disjoint
  E1, E2, E1 U E2 in CC
- finitely additive:
  y(U_1..n Ek) = sum_1..n y(Ek)
  for every disjoint finite sequence Ek
  sequence (Ek) in CC, U_1..n Ek in CC
- countably additive:
  y(U En) = sum y(En)
  for every disjoint sequence En
  sequence (Ek) in CC, U Ek in CC
- subadditive:
  y(E1 U E2) <= y(E1) + y(E2)
  E1, E2, E1 U E2 in CC
- finitely subadditive:
  y(U_1..n Ek) <= sum_1..n y(Ek)
  for every finite sequence Ek
  sequence (Ek) in CC, U_1..n Ek in CC
- countably subadditive:
  y(U En) <= sum y(En)
  for every sequence En
  sequence (Ek) in CC, U Ek in CC

Note that although we require the final union to be
present in CC, we don't require that
intermediate unions from 2 to n-1 be present in CC.

observation 1.20
y, CC as before.
nullset in CC
y(nullset) = 0
- if y countably additive on CC, then
  it is finitely additive.

- if y is countably subadditive on CC, then
it is finitely subadditive on CC.
pf:
- y countably additive.
(Ek : k=1..n) disjoint and U_1..n Ek in CC
make infinite sequence Fk by appending nullset
nullset in CC, U Fk = U Ek in CC, so disjoint seq in CC
y(U_1..n Ek) = y(U Fk)   [countable additivity]
= sum y(Fk) = sum_1..n y(Ek)
thus y finitely additive.
similarly, y finitely subadditive.

[basically we can "weaken" from countable to finite
by appending nullset/zero to the union/sum]

Lem 1.21
(En) sequence in algebra AA
there exists disjoint sequence (Fn) in AA st
1) U_1..N En = U_1..N Fn for every N
2) U En = U Fn
and if AA is sigma-algebra, it includes this.
pf:
let Fn = En \ (E1 U ... U En-1) [be the "new content"]
AA includes this since it's algebra.
1)
induction:
base case: F1 = E1
ind step: stmt valid for N: U_1..N Fn = U_1..N En
U_1..N+1 Fn
(U_1..N Fn) U F_N+1
(U_1..N En) U (E_N+1 \ U_1..N En)
A U (B \ A)
A U (B int Ac)
A U B
U_1..N+1 En
2)
say x in U En
x in En for some n
x in U_1..n Ek
x in U_1..n Fk
x in U Fn

similarly, if x in U Fn, x in U En

show Fn disjoint:

```
        take Fn, Fm, n<m
        Fm = Em \ (E1 U .. U Em-1)
        U_1..m-1 Ek = U_1..m-1 Fk
            Fn is subset of this since n<m
            thus Fn int Fm = nullset, since we have:
            Fm = Em \ (... Fn ...)

    [basically we make a sequence into just disjoint pieces]

Lem 1.22
    y as before on algebra AA
    1) y additive ->
        y finitely additive, monotone, finitely subadditive
    2) y countable additive ->
        y countably subadditive
    pf:
    1) y additive.
    - finitely additive:
        (Ek : k=1..n) disjoint finite seq in AA
        the partial unions
            U_1..k Ei for k=1..n are in AA
        U_1..n-1 Ek  and  En  disjoint, so
            y(U_1..n Ek) = y(U_1..n-1 Ek) + y(En)
            repeat this argument:
            y(U_1..n Ek) = sum_1..n y(Ek)

        [basically if it works for 2, it works for n]

    - monotonicity:
        E1, E2 in AA, E1  subset  E2
        E1, (E2 \ E1) in AA
        E1 int (E2 \ E1) = nullset  disjoint
        E1 U (E2 \ E1) = E2 in AA so
            y(E1) + y(E2 \ E1) = y(E2)
            y(E2) - y(E1) = y(E2 \ E1) >= 0
                since y non-neg extended
            y(E2) >= y(E1)

    - finite subadditivity:
        (Ek : K=1..n) finite seq in AA
        let Fk be the "new content"
        y(U_1..n Ek)
        y(U_1..n Fk)            [construction]
        sum_1..n Fk             [finite additivity]
        <= sum_1..n y(Ek)       [monotonicity]
```

    2) y countably additive.
        (En) infintie seq in AA, U En in AA
            [even though it's only an algebra]
        (Fn) new content
        y(U En)
        y(U Fn)   [construction]
        sum y(Fn)   [countable additivity]
        <= sum y(En)   [monotonicity]
            [i guess the reasoning is that for each n,
            Fn subset En, thus y(Fn) <= y(En), thus
            sum y(Fn) <= sum(En)]

[REMEMBER: additivity always implies disjoint.
subadditivity is for any sets]

Prop 1.23
    y as before on algebra AA
    - y additive + countably subadditive
        -> y countably additive
    pf:
    y additive and countably subadditive.
        -> monotonicity
        -> finite additivity
    (En) disjoint seq in AA, U En in AA
    for each N,
        y(U_1..N En) <= y(U En)   [monotonicity]
        sum_1..N y(En) <= y(U En)   [finite additivity]
            since this holds for every N,
        sum y(En) <= y(U En)
    on the other hand, by countable subadditivity.
        y(U En) <= sum y(En)
        thus they are equal, and y is countably additive.

AA sigma-algebra. mu on AA is a MEASURE:
    - mu in [0,inf]
    - mu(nullset) = 0
    - countable additivity: En disjoint, mu(U En) = sum mu(En)

Lem 1.25
    measure mu on sigma-algebra AA has properties:
    1) finite additivity
    2) monotonicity
    3) E1,E2 in AA
        E1 subset E2

```
    mu(E1) < inf
    then mu(E2 \ E1) = mu(E2) - mu(E1)
4) countable subadditivity
5) finite subadditivity
pf:
1) countable additivity -> finite additivity        [1.20]
2) finite additivity -> additivity -> monotonicity  [1.22]
3) E1,E2 in AA, E1  subset  E2,
    E1, (E2 \ E1) disjoint
    E1 U (E2 \ E1) = E2
    mu(E2) = mu(E1) + mu(E2 \ E1)         [additivity]
    mu(E2) - mu(E1) = mu(E2 \ E1)         [if mu(E1) < inf]
4) countable additivity -> countable subadditiity   [1.22]
5) countable subadditivity -> finite subadditivity  [1.20]
```

Measures of a Sequence of Sets

```
Thm 1.26 MONOTONE CONVERGENCE THM FOR SEQUENCE OF MSBL SETS
    mu measure on sigma-algebra AA
    (En) monotone sequence in AA
    - if En ^, lim mu(En) = mu(lim En)
    - if En v, lim mu(En) = mu(lim En)
        if there exists set A in AA st
        mu(A) < inf, and E1  subset  A
        [if sequence start is finite]
    pf:
    - if En ^, lim En = U En
    - if En v, lim En = int En
    - En monotone -> mu(En) monotone by [monotonicity]
        thus lim mu(En) exists in [0,inf]
    1)
        En ^
        mu(En) ^
        - if mu(En0) = inf for some n0,
            lim mu(En) = inf
            En0  subset  U En = lim En
            inf = mu(En0) <= mu(lim En)
            thus lim mu(En) = inf = mu(lim En)
        - mu(En) < inf foreach n
            [since mu(En) increasing, it either becomes
            infinite at some point, as in the case above,
            or it never does]
            let Fn be new content
                define E0 := nullset
                since En increasing, Fn = En \ En-1
```

```
    mu(lim En)
    mu(U En)
    mu(U Fn)
    sum mu(Fn)                [countable additivity]
    sum mu(En \ En-1)
    sum [mu(En) - mu(En-1)]   [1.25]
        sum of series is limit of sequence
        of partial sums.
    lim sum_1..n [mu(Ek) - mu(Ek-1)]
    lim mu(En) - mu(E0)
    lim mu(En)
    thus mu(lim En) = lim mu(En)
- if En v, and E1 contained in finite set.
    [if it's decreasing but always infinite,
    limit is infinity. if at some point it becomes
    finite, then cut off the infinite part]
    let Fn be new content
        [Fn := En \ En+1 this time]

    WTS: E1 \ (int En) = U Fn
        [LHS: everything but the limit
        RHS: all new content]
        ->
        x in E1 \ (int En)
        x in E1, and x not in every En (not in limit?)
        since En decreasing, there exists n0 st
            x not in En0+1, and
            x not in any further set
        x in En0 \ En0+1
        x in Fn0  subset  U Fn
        <-
        x in U Fn
        x in Fn0 for some n0
        x in En0 \ En0+1
        x in En0  subset  E1
        x not in En0+1
        x not in int En
        thus x in E1 \ (int En)
        [this is a little weird because usually we
        start off with union from 1 to n, and show
        that U En = U Fn forall n, and at the limit
        as well using a "both subset" argument.
        but here we are removing the next thing,
        so we actually never include the limit.
        ok maybe it's not that strange if we just
```

```
            consider union 1..n and note that U Fn
            does not include the limit]

        mu(E1 \ (int En)) = mu(U Fn)
            mu(int En) <= mu(E1) < inf  [given]
        mu(E1 \ (int En))
        mu(E1) - mu(int En)
        mu(E1) - mu(lim En)

        mu(U Fn)
        sum mu(Fn)      [countable additivity]
        sum mu(En \ En+1)
        sum [mu(En) - mu(En+1)]
        lim sum_1..n [mu(Ek) - mu(Ek+1)]
        lim [mu(E1) - mu(En+1)]
        mu(E1) - lim mu(En)

        combining results:
        mu(E1) - mu(lim En) = mu(E1) - lim mu(En)
        mu lim(En) = lim mu(En)

Remark 1.27
    particular cases for decreasing sequence
    lim mu(En) = mu(lim En) if any is satisfied:
    - mu(X) < inf
    - mu(E1) < inf
    - mu(En0) < inf for some n0
    pf:
    1) E1  subset  X  thus mu(E1) finite
    2) mu(E1) finite
    3)
        mu(En0) < inf for some n0
        define Fn by dropping first n0 terms of En
            Fn = E_n0+n
            they have the same limsup and liminf
            [because if event happens i.o. in
            original sequence, it will still
            happen i.o. in the new sequence.
            same for all but finitely]
            lim Fn = lim En
        Fn decreasing
        Fn  subset  En0  forall n
            mu(En0) < inf
            apply MCT for set sequence
            lim mu(Fn) = mu(lim Fn) = mu(lim En)
```

```
        since the numerical sequence mu(Fn)
            is obtained by dropping first n0 terms from mu(En),
            lim mu(Fn) = lim mu(En)
        lim mu(En) = mu(lim En)

measure mu on sigma-algebra AA
    arbitrary sequence (En)
        liminf and limsup exist in AA
        mu(liminf En), mu(limsup En) are defined
    mu(En) is numerical sequence in [0,inf]
        liminf mu(En) := lim_n inf_k>n mu(Ek)
        limsup mu(En) := lim_n sup_k>n mu(Ek)
        exist in [0,inf]
        [this is because inf_k>n is an increasing seq,
        and sup_k>n is a decreasing seq, and monotone
        number sequences always have limits]

Thm 1.28
    mu measure on sigma-algebra AA
    a)
            mu(liminf En) <= liminf mu(En)
    b)
        mu(En) finite
            limsup mu(En) <= mu(limsup En)
    c)
        [special case of 1]
        if lim En, lim mu(En) exist:
            mu(lim En) <= lim mu(En)
    d)
        if lim En exists, mu(En) finite:
            lim mu(En) exists
            mu(lim En) = lim mu(En)
    pf:
    1)
        mu(liminf En)
        mu(lim_n int_k>n Ek)
        lim_n mu(int_k>n Ek)          [MCT: int_k>n Ek ^]
        liminf mu(int_k>n Ek)         [limit = liminf]
        <= liminf mu(En)              [int_k>n Ek  subset  En]
    2)
        mu(En) finite

        mu(limsup En)
        mu(lim_n U_k>n Ek)
        lim_n mu(U_k>n Ek)     [MCT: U_k>n Ek v, finite measure]
```

```
        limsup mu(U_k>n Ek)            [limit = limsup]
        >= limsup mu(En)               [En  subset  U_k>n Ek]
    3)
        lim En, lim mu(En) exist
        from (1)
            mu(liminf En) <= liminf mu(En)
            mu(lim En) <= lim mu(En)        [limit = liminf]
    4)
        lim En exists
        mu(En) finite

        limsup mu(En) <= mu(limsup En) (2) [mu(En) finite]
        mu(liminf En)             [lim exists: limsup = liminf]
        <= liminf mu(En)          (1)
        thus lim mu(En) exists    [liminf <= limsup always]

        equals mu(lim En)         [sandwich]
```

Measurable Space and Measure Space

```
AA sigma-algebra of X
    pair (X, AA) is MEASURABLE SPACE
        subset E of X is AA-measurable if E in AA

    mu measure on AA
        triple (X, AA, mu) is MEASURE SPACE

    mu is finite if mu(X) < inf
        then (X, AA, mu) is FINITE MEASURE SPACE

    mu is SIGMA-FINITE if
        there exists (En) st U En = X
        and mu(En) < inf for each n
        (X, AA, mu) is SIGMA-FINITE MEASURE SPACE

    set D in AA is SIGMA-FINITE SET if
        there exists sequence (Dn) st U Dn = D
        and mu(Dn) < inf for each n

Lem 1.31
    1)
        (X, AA, mu)
        sigma-finite set D in AA
        then there exists increasing (Fn)
            lim Fn = D
```

```
            mu(Fn) < inf
        there exists disjoint (Gn)
            U Gn = D
            mu(Gn) < inf
    2)
        (X, AA, mu) sigma-finite measure space
        every D in AA is a sigma-finite set
    pf:
    1)
        D sigma-finite set
        let Fn = U_1..n Dk
        lim Fn
        U Fn     [increasing]
        U Dn
        D

        mu(Fn)
        mu(U_1..n Dk)
        <= sum_1..n mu(Dk)
        < inf

        let Gn be new content of Fn
            Gn disjoint
            U Gn = U Fn = D

        mu(Gn) <= mu(Fn) < inf

    2)
        (X, AA, mu) sigma-finite
            there exists (En) st U En = X, mu(En) < inf
        D in AA
        let Dn = D int En
            U Dn = D
            mu(Dn) <= mu(En) < inf
        D is sigma-finite set

mu on sigma-algebra AA
    subset E of X is a NULL SET wrt mu:
        - mu(E) = 0

observation 1.33
    countable union of null sets is a null set
    pf:
    (En) sequence of null sets
    mu(U En) <= sum mu(En) = 0     [countable subadditivity]
```

```
mu on sigma-algebra AA
    AA is COMPLETE wrt mu if
        it contains all subsets of its nullsets

    when AA complete wrt mu, (X, AA, mu) is COMPLETE MEASURE SPACE

    example:
    X = {a,b,c}
    AA = {nullset, {a}, {b,c}, X} is sigma-algebra
    mu(nullset) = 0
    mu({a}) = 1
    mu({b,c}) = 0
    mu(X) = 1
    not complete because we dont have subsets of {b,c}

measurable space (X, AA)
    E is ATOM of the measurable space:
        nullset and E are the only AA-measurable subsets of E
        [it does not have subsets in AA]

measure space (X, AA, mu)
    E is ATOM of the measure space:
        - mu(E) > 0
        - E0  subset  E, E0 in AA
            -> mu(E0) = 0  or  mu(E0) = mu(E)
        [subsets have either same (poz) or zero measure]

    example:
    (X, AA)
    X = {a,b,c}
    AA = {nullset, {a}, {b,c}, X}
    mu(nullset) = 0
    mu({a}) = 1
    mu({b,c}) = 2
    mu(X) = 3
        {b,c} is an atom of the measure space
        because its subsets in AA have:
            mu({b,c}) = 2 > 0
            mu(nullset) = 0
        it is also an atom of the measurable space
        since only nullset, {b,c} are its subsets in AA
```

Measurable Mapping

```
f: (D  subset  X) -> Y
    DD(f) := domain of f = D  subset  X
    RR(f) := range of f =
        {y in Y : y = f(x) for some x in DD(f)}  subset  Y

    image of DD(f) by f:
        f( DD(f) ) = RR(f)

    E  subset  Y
    preimage of E under f:
        f^-1(E) := {x in X : f(x) in E}
                 = {x in DD(f) : f(x) in E}
                 [note: X \ DD(f) will be points on which
                 f is not defined, so we won't consider them]
        E is arbitrary subset of Y,
        need not be subset of RR(f),
        and may be disjoint from RR(f),
        in which case f^-1(E) = nullset

        f(f^-1(E))  subset  E
        [once again, preimage of E is points that get mapped
        to E. so image of the preimage should be exactly E,
        unless E contains points that are not mapped to,
        and thus have no preimage. in that case subset.]

observation 1.36
    f: (DD(f)  subset X) -> (RR(f)  subset  Y)
    E, E_a arbitrary subsets of Y
    1)
        f^-1(Y) = DD(f)
        {x in DD(f) : f(x) in Y} -> all of DD(f)
    2)
        f^-1(Ec)
        f^-1(Y \ E)
        f^-1(Y) \ f^-1(E)
        DD(f) \ f^-1(E)
    3)
        f^-1(Ec) = (f^-1(E))c   provided DD(f) = Y
        [is this a mistake? this is true if DD(f) = Y,
        but i think its true if DD(f) = X because then
        (f^-1(E))c = X \ f^-1(E)  and since we can tell
        that f^-1(E) is in X, the complement should be
        relative to X.]
        [OKAY, if we specialize (2) with DD(f) = X, it
        is in fact true. It is also true if DD(f) = X = Y
```

```
        so that is an even more specific case.]
    4)
        f^-1(U Ea) = U f^-1(Ea)
    5)
        f^-1(int Ea) = int f^-1(Ea)

Prop 1.37
    f: (DD(f)  subset  X) -> (RR(f)  subset Y)
    BB sigma-algebra of Y
    -> f^-1(BB) sigma-algebra of DD(f)

    [at first I was confused: why is BB sigma-algebra of Y,
    and not of RR(f)? its because BB is a sigma-algebra of Y,
    so (BB int RR(f)) is a sigma-algebra of RR(f). for example,
    Y = R, BB = Borel, RR(f) = {0,1}, then (Borel int RR(f)) =
    {nullset, {0}, {1}, {0,1}} and its preimage is also sig-alg
    its just that f^-1(BB) is nullset for the rest of the sets]

    pf:
    1)
        Y in BB
        f^-1(Y) = DD(f)                     [1.36]
        thus DD(f) in f^-1(BB)
    2)
        A in f^-1(BB)
        WTS: DD(f) \ A
        DD(f) \ f^-1(B)   [A = f^-1(B) for some B in BB]
        f^-1(Bc)                             [1.36]
            Bc in BB
            thus f^-1(Bc) in f^-1(BB)
    3)
        (An) in f^-1(BB)
        An = f^-1(Bn) for some Bn in BB
        U Bn in BB
        f^-1(U Bn) in f^-1(BB)
        U f^-1(Bn) in f^-1(BB)
        U An in f^-1(BB)

two measurable spaces (X,AA) and (Y,BB)
f: (DD(f)  subset  X) -> (RR(f)  subset  Y)
    f is AA/BB-measurable mapping if
        f^-1(B) in AA for every B in BB,
        meaning f^-1(BB)  subset  AA

        [f maps sets in AA to sets in BB
```

```
        f maps measurable sets to measurable sets]

        we know that f^-1(BB) is a sigma-algebra of DD(f)
        so to be AA/BB-measurable,
            we need f^-1(BB)  subset  AA
                [AA includes preimage sigma-algebra]
            also since Y in BB,
            we need f^-1(Y) = DD(f) in AA
                [domain in AA]
                to construct AA/BB-measurable f on
                D  subset  X, we must have D in AA
                [if we want to restrict f to some subset,
                that subset must be in AA]

observation 1.39
    (X, AA) (Y, BB)
    f is AA/BB measurable
    - if AA1 is sigma-algebra of X st
        AA  subset  AA1,
        then f is AA1/BB-measurable
    - if BB0 is sigma-algebra of Y st
        BB0  subset  BB,
        then f is AA/BB0-measurable
    pf:
    1)
        f^-1(BB)  subset  AA  subset  AA1
    2)
        f^-1(BB0)  subset  f^-1(BB)  subset  AA

Thm 1.40 CHAIN RULE FOR MEASURABLE MAPPINGS
    (X, AA) (Y,BB) (Z,CC)
    f: (DD(f)  subset  X) -> (RR(f)  subset  Y)
    g: (DD(g)  subset  Y) -> (RR(g)  subset  Z)
        also RR(f)  subset  DD(g)
        thus (g o f) defined with
            DD(g o f)  subset  X
            RR(g o f)  subset  Z
    if f is AA/BB measurable,
    and g is BB/CC measurable,
    then (g o f) is AA/CC measurable
    pf:
        know:
        f^-1(BB)  subset  AA
        g^-1(CC)  subset  BB
    f^-1(g^-1(CC))  subset  f^-1(BB)  subset  AA
```

```
[to check measurability,
instead of checking preimage of BB is subset of AA,
we can check preimage of generating set subset of AA
(plus domain in AA)]

Thm 1.41
    (X, AA) (Y,BB)
        BB = sigma(CC)
        CC arbitrary collection of subsets of Y
    f: (DD(f) in AA) -> (RR(f)  subset  Y)
    -> f is AA/BB-measurable map iff
        f^-1(CC)  subset  AA

        [note that we changed f's domain from being subset of X
        to being in AA.
        CC may not contain Y,
        but we require DD(f) in AA
        "to construct a AA/BB measurable map f on subset D of X,
        we must assume D in AA"
        so this is an extra condition to require domain in AA.]

    pf:
    ->
    CC  subset  sigma(CC) = BB
    f^-1(CC)  subset  f^-1(BB)  subset  AA   [f AA/BB-measurable]
    <-
    f^-1(CC)  subset  AA
    sigma(f^-1(CC))  subset  sigma(AA) = AA
    f^-1(sigma(CC))  subset  AA                     [1.14]
    f^-1(BB)  subset  AA

Prop 1.42
    (X,AA) (Y,BB_y)
        Y is topological space
        BB_y Borel sigma-algebra of Y
    f: (DD(f) in AA) -> (RR(f)  subset  Y)
    OO_y, CC_y collection of all open/closed sets in Y
    - f ia AA/BB_y-measurable iff f^-1(OO_y)  subset  AA
    - f ia AA/BB_y-measurable iff f^-1(CC_y)  subset  AA
    pf:
    BB_y = sigma(OO_y) = sigma(CC_y)
    this is particular case of (1.41)

    [check that preimage of generating set (open/closed) in AA
```

```
    to get measurability: f: AA -> Borel]

Thm 1.43
    (X,BB_x) (Y,BB_y)
        X, Y topological spaces
        BB_x, BB_y Borel sigma-algebras of X,Y
    f continuous, defined on D in BB_x,
        then f is BB_x/BB_y-measurable
    pf:
    V open set in Y
    f continuous on D: maps open sets to open sets
        f^-1(V) = U int D for open set U in X
        thus f^-1(V) in BB_x
    this holds for every open set
        by (1.42), f is BB_x/BB_y-measurable

        [note we were given DD(f) in AA,
        f defined on D in BB_x]

    particular case:
    real-valued continuous f
        defined on D in BB_x
        (Y,BB_y) = (R,BB_R)
        by (1.43) f is BB_x/BB_R-measurable
```

Induction of Measure by Measurable Mapping

```
mu measure on sigma-algebra AA
    a measurable map:(X,AA)->(Y,BB) induces a measure on BB,
    called IMAGE MEASURE induces by the map.

Thm 1.44 IMAGE MEASURE
    (X,AA) (Y,BB)
    f AA/BB-measurable: X -> Y
    mu measure on AA
    set function v(B) := mu(f^-1(B)) for B in BB
        is a measure on BB

    pf:
    1)
        know f^-1(BB)  subset  AA
        v(B) = mu(f^-1(B)) in [0,inf]
    2)
        v(nullset) = mu(f^-1(nullset)) = mu(nullset) = 0
    3)
```

```
    (Bn) disjoint sequence in BB
    v(U Bn)
    mu(f^-1(U Bn))          [def]
    mu(U f^-1(Bn))          [set theory]
    sum mu(f^-1(Bn))        [f^-1(Bn) disjoint sequence in AA]
    sum v(Bn)
```

Problems

```
1.1
    (En), (Fn) sequences
    part 1
    a)
        WTS: liminf En U liminf Fn  subset  liminf (En U Fn)

        x in LHS

        x in liminf En
            or x in liminf Fn

        x in U_n int_k>n Ek
            or x in U_n int_k>n Fk

        there exists n st x in int_k>n Ek
            or there exists m st x in int_j>m Fj

        there exists n st x in Ek forall k>n
            or there exists m st x in Fj forall j>m

        there exists n st x in (Ek U Fk) forall k>n
            or there exists m st x in (Ej U Fj) forall j>m

        we see that in either case, there exists a number,
            (num = n or m), such that x in (Ek U Fk) forall k>num

        there exists a number st x in int_k>num (Ek U Fk)

        x in U_num int_k>num (Ek U Fk)

        x in liminf (En U Fn)

    b)
        WTS: liminf (En U Fn)  subset  liminf En U limsup Fn

        x in LHS
```

```
    x in liminf (En U Fn)
    x in U_n int_k>n (Ek U Fk)
    there exists n st x in int_k>n (Ek U Fk)
    there exists n st x in (Ek U Fk) forall k>n
    there exists n st forall k>n, x in either Ek or Fk or both

    consider sets (Ek U Fk) for k>n.
    foreach k, x in Ek, Fk, or both

    case 1:
        x in Ek a finite number of times
        x in Fk infinite number of times
        x in liminf Fn
        x in limsup Fn
        x in RHS

    case 2:
        x in Fk a finite number of times
        x in Ek infintie number of times
        x in liminf En
        x in RHS

    case 3:
        x appears infinite number of times in both Ek,Fk
        x in limsup En, and x in limsup Fn
        x in (limsup En) int (limsup Fn)
            subset  limsup Fn
            subset  RHS

c)
    WTS: liminf En U limsup Fn  subset  limsup (En U Fn)

    x in LHS

    if x in liminf En,
        x in limsup En
        x in limsup (En U Fn)
    if x in limsup Fn,
        x in limsup (En U Fn)

d)
    WTS: limsup (En U Fn)  subset  limsup En U limsup Fn

    x in LHS
    x in int_n U_k>n (Ek U Fk)
```

```
    foreach n, for some k>n, x in (Ek U Fk)
    foreach n, for some k>n, x in Ek, Fk, or both

    case 1:
        x in Ek finite number of times
        x in Fk infinite number of times
        x in limsup Fn
        x in RHS

    case 2:
        x in Fk finite number of times
        x in Ek infinite number of times
        x in limsup Ek
        x in RHS

    case 3:
        x in Ek infinitely often,
            and x in Fk infinitely often
        x in limsup Ek
            and x in limsup Fk
        x in (limsup Ek) int (limsup Fk)
            subset  (limsup En)
            subset RHS

part 2
a)
    WTS: liminf En int liminf Fn  subset  liminf (En int Fn)

    x in LHS

    x in liminf En
        and x in liminf Fn

    there exists n st x in Ek for k>n
        and there exists m st x in Fj for j>m

    n0 = max(m,n) then x in Ek and Fk for k>n0

    x in (Ek int Fk) for k>n0

    x in int_k>n0 (Ek int Fk)

    x in U_n int_k>n (Ek int Fk)

    x in liminf (En int Fn)
```

b)

```
WTS: liminf (En int Fn)  subset  liminf En int limsup Fn

x in LHS

x in liminf (En int Fn)

x in (En int Fn) infinitely often

x in En infinitely often,
    and x in Fn infinitely often

x in liminf En,
    and x in liminf Fn

x in liminf En,
    and x in limsup Fn

x in RHS
```

c)

```
WTS: liminf En int limsup Fn  subset  limsup (En int Fn)

x in LHS

x in liminf En
    and x in limsup Fn

there exists n st x in Ek forall k>n
    and forall m, there exists j>m st x in Fj

if we choose m>n, there exists j>m st x in Fj but also
    x in Ej as well, so x in (Ej int Fj)
    let's call this value of m0

so for the first n numbers, choose m0,
and for all numbers greater than n,
    we can find a further j st x in (Ej int Fj)

thus for all numbers we can find j st x in (Ej int Fj)

for all numbers n, x in U_k>n (Ek int Fk)

x in int_n U_k>n (Ek int Fk)
```

```
    x in limsup (En int Fn)

d)
    WTS: limsup (En int Fn)  subset  limsup En int lmisup Fn

    x in LHS

    x occurs in (En int Fn) infinitely often
        (En int Fn)  subset  En
        (En int Fn)  subset  Fn

    x occurs in En infinitely often,
        and x occurs in Fn infinitely often

    x in limsup En,
        and x in limsup Fn

    x in RHS

part 3
lim En, lim Fn exist

WTS: lim (En U Fn) exists
    WTS: liminf (En U Fn) = limsup (En U Fn)
    ->
    always
    <-
    x in limsup (En U Fn)
    x in (limsup En) U (limsup Fn)   (showed, subset)
    x in (liminf En) U (liminf Fn)   (lim exists)
    x in liminf (En U Fn)            (showed, subset)

WTS: lim (En int Fn) exists
    WTS: liminf (En int Fn) = limsup (En int Fn)
    ->
    always
    <-
    x in limsup (En int Fn)
    x in limsup En int limsup Fn     (showed, subset)
    x in liminf En int liminf Fn     (lim exists)
    x in liminf (En int Fn)          (showed, subset)

WTS: lim (En U Fn) = lim En U lim Fn
    ->
```

```
        x in LHS
        x in liminf (En U Fn)            (showed lim exists)
        x in (limsup En) U (limsup Fn)   (showed, subset)
        x in (lim En) U (lim Fn)         (lim exists)
        <-
        x in RHS
        x in (lim En) U (lim Fn)
        x in (liminf En) U (liminf Fn)   (lim exists)
        x in liminf (En U Fn)            (showed, subset)
        x in lim (En U Fn)               (showed lim exists)

    WTS: lim (En int Fn) = lim En int lim Fn
        ->
        x in LHS
        x in lim (En int Fn)
        x in liminf (En int Fn)          (showed lim exists)
        x in (limsup En) int (limsup Fn) (showed, subset)
        x in (lim En) int (lim Fn)       (lim exists)
        <-
        x in RHS
        x in (lim En) int (lim Fn)
        x in (liminf En) int (liminf Fn) (lim exists)
        x in liminf (En int Fn)          (showed, subset)
        x in lim (En int Fn)             (showed lim exists)

1.2
    a)
    (An) sequence
    (Bn) sequence got by dropping finite # of terms of An

    let ld be the index of the last term dropped
    let d < inf be the number of terms dropped

                                | ld
    An: *************************123456789************************
    Bn: *____*__**__***___**__**_123456789************************
    Bn: ***********123456789************************
                  |- point after which Bk = Ak+d
              (ld-d+1)

    number of terms remaining: ld-d
    starting at (ld-d+1), Bk = Ak+d

    WTS: liminf Bn = liminf An
        ->
```

```
x in liminf Bn
there exists n, st x in Bk forall k>n

    case 1:
        n >= (ld-d+1)
        x in Bk forall k >= n
            this is past the threshold where Bk = Ak+d, so
        x in Aj forall j >= (n+d)
        so there exists number (n+d) st
            x in Aj forall j >= (n+d)
        thus x in liminf An

    case 2:
        n < (ld-d+1)
        x in Bk forall k >= n
            add (ld-d+1) so we can be past threshold
            where Bk = Ak+d
            x is still in those sets since they are > n
        x in Bk forall k >= (n+ld-d+1)
            this is past the threshold where Bk = Ak+d, so
        x in Aj forall j >= (n+ld-d+1+d) = (n+ld+1)
        so there exists number (n+ld+1) st
            x in Aj forall j >= (n+ld+1)
        thus x in liminf An

<-
x in liminf An
there exists n, st x in Ak forall k>n

    case 1:
        n >= (ld+1)
            then Bk = Ak+d
        x in Ak forall k >= n
        x in Bj forall j >= (n-d)
        thus there exists number (n-d) st
            x in Bj forall j >= (n-d)
        thus x in liminf Bn

    case 2:
        n < (ld+1)
            add (ld+1) to get past threshold where Bk = Ak+d
        x in Ak forall k >= n
        x in Ak forall k >= (n+ld+1)
            this is past threshold where Bk = Ak+d, so
        x in Bj forall j >= (n+ld+1-d)
```

```
            thus there exists number (n+ld+1-d) st
                x in Bj forall j >= (n+ld+1-d)
            thus x in liminf Bn

                            | ld
An: *************************123456789************************
Bn: *____*__**__***___**__**_123456789************************
Bn: ***********123456789************************
            |- point after which Bk = Ak+d
            (ld-d+1)


WTS: limsup Bn = limsup An
    ->
    x in limsup Bn
    foreach n, there exists k >= n, st x in Bk
        consider sequence k1,k2,k3...
    note that k_(ld-d+1) will suffice for:
        B_(ld-d+1)            [its the corresponding term]
        A_(ld-d+1+d)          [this is the same set]
            [because we are past the threshold where Bk = Ak+d]
        Ai for i < (ld-d+1+d) [ki will suffice for the first i terms]
    thus we can choose k_(ld-d+1) to be the k's corresponding
        to the first (ld+1) terms of An.
    for the k's corresponding to the An's for n > (ld+1),
        just choose the (n-d)th term from the sequence of k's for Bn.
        since Bk = Ak+d at this point, the k's will suffice.
    this way for each n we have found k>n st x in Ak
    thus x in limsup An

    <-
    x in limsup An
    foreach n, there exists k>n st x in Ak
        consider sequence k1,k2,k3...
        note k_(ld+1) suffices for A_(ld+1)
            also suffices for B_(ld-d+1)
                [Bk = Ak+d starting at this point]
            also suffices for Bi for i < (ld-d+1)
    create new sequence of j's for Bn (to serve as k's for An)
        it equals k_(ld+1) for the first (ld-d+1) terms
        then set it equal to k_i for i > (ld+1)
    this way, for each n, we found a number j > n st x in Bj
    thus x in limsup Bn
```

```
WTS: lim Bn exists <=> lim An exists
    ->
    if lim Bn exists, liminf Bn = limsup Bn
    we showed that liminf Bn = liminf An
        and limsup Bn = limsup An
        thus liminf An = limsup An
        thus lim An exists
    <-
    if lim An exists, liminf An = limsup An
    we showed that liminf An = liminf Bn
        and limsup An = limsup Bn
        thus liminf Bn = limsup Bn
        thus lim Bn exists

WTS: lim An, lim Bn exist, they are equal
    lim An = liminf An = liminf Bn = lim Bn


b)
(An) (Bn)
An = Bn for all but finitely many n

there exists n0 st Ak = Bk forall k > n0
consider sequence Cn, created by dropping the first n0 terms of An
    note we get the same sequence if we drop first n0 terms of Bn

applying (part a) to (An) and (Cn), we get
    liminf An = liminf Cn
    limsup An = limsup Cn
applying (part a) to (Bn) and (Cn), we get
    liminf Bn = liminf Cn = liminf An
    limsup Bn = limsup Cn = limsup An

WTS: lim Bn exists <=> lim An exists
    ->
    lim Bn exists
    liminf Bn = limsup Bn
    liminf An = liminf Bn = limsup Bn = limsup An
    lim An exists
    <-
    lim An exists
    liminf An = limsup An
    liminf Bn = liminf An = limsup An = limsup Bn
    lim Bn exists

WTS: lim Bn, lim An exist, then they are equal
```

```
        lim Bn = liminf Bn = liminf An = lim An

1.3
    (En) disjoint sequence

    WTS: lim En exists
        liminf En = limsup En
        ->
        x in liminf En
        x in U_n int_k>n Ek
            Ek disjoint, so int_k>n nullset foreach n
        x in U_n nullset
        x in nullset
        <-
        x in limsup En
        x in int_n U_k>n Ek
        for each n,
            U_k>n Ek  will not contain  Ei for i=1..(n-1)
                [since they are disjoint]
            thus for each n, En will not be in limsup.
            thus limsup is empty = nullset

        additionally, if x in limsup, x occurs i.o.
        but if x in En, we know it never occurs again
        since the En are disjoint.

1.4
    a in R
    (xn) sequence of points in R, distinct from a, st lim xn = a

    WTS: lim {xn} exists
        liminf {xn} = limsup {xn}
        ->
        x in liminf {xn}
        x in U_n int_k>n {xk}
        there exists n, st x in int_k>n {xk}
            case 1:
                xk are all the same. then lim xk = a, xk = a forall k
                we know this is not the case since xk distinct from a
            case 2:
                xk are not all the same. then int_k>n {xk} = nullset
        x in U_n nullset
        x in nullset
        <-
        x in limsup {xn}
```

```
    x in int_n U_k>n {xk}
    foreach n,
        U_k>n {xk} will not contain xi for i=1..(n-1)

        each xi can either (be or not be a) and also
            each xi can (occur i.o or finite # of times)

        case 1:
            xi occurs infinitely often, xi = a
            impossible, xi =/= a

        case 2:
            xi occurs finite number of times, xi = a
            impossible, xi =/= a

        case 3:
            xi occurs infinitely often, xi =/= a
            impossible, because then limit =/= a

        case 4:
            xi occurs finitely often, xi =/= a
            this is the only remaining choice.
            since xi is arbitrary, we know that each xi
            occurs a finite number of times.

            thus U_k>n {xk} = nullset
            since for each xi, we can find n0 large enough
            so that it is not in U_k>n0 {xk}
            and thus not in their intersection.
            thus limsup empty.

WTS: lim {xn} = nullset
    limsup {xn} = nullset = liminf {xn}
    thus limit exists

1.5
    E subset of R
    t in R
    E + t = {x + t : x in E}
        translate E by t
    (tn) strictly decreasing sequence in R
        lim tn = 0
    En = E + tn

    a) E = (-inf, 0)
```

```
    En = {x + tn : x in (-inf, 0)}
    En = (-inf, tn)

    liminf En
    U_n int_k>n (-inf, tk)
        consider int_k>n (-inf, tk)

        does the intersection contain positive points?
        assume it contains e > 0.
        but for e > 0, we know there exists N
            st 0 < tk < e for k>N           [lim = 0, dec]
            thus int_k>n (-inf, tk) does not contain e.
        since e arbitrary, it contains no positive points.

        does the intersection contain 0?
        tk =/= 0 because if tk = 0, and it's
            strictly decreasing, further tk are < 0,
            so lim =/= 0, contra. thus tk > 0, forall k
            thus 0 in (-inf, tk) forall k

        int_k>n (-inf, tk) = (-inf, 0] forall n

        U_n int_k>n (-inf, tk)
        U_n (-inf, 0]
        (-inf, 0]

    limsup En
    int_n U_k>n (-inf, tk)
        (-inf, tk) is decreasing, so union is
        U_k>n (-inf, tk) = (-inf, tn)

        int_n (-inf, tn)
        once again:
            - this inresection contains no positive pts,
                because we can find 0 < tn < e and so
                (-inf, tn) will not include e.
            - zero is in here because tn > 0 forall n,
                thus zero in (-inf, tn) forall n.
        (-inf, 0]

    thus limsup En = liminf En = (-inf, 0] = lim En

b)
    E = {a}
    En = {x + tn : x in E}
```

```
        = {a + tn}
    since a in R, (a + tn) seq in R,
        distinct from a,
        lim (a + tn) = a
        by (1.4) we know lim {a + tn} exists and = nullset

c)
    E = [a,b]
        a < b
    En = [a + tn, b + tn]

    liminf En
    U_n int_k>n [a + tk, b + tk]
    consider the liminf.
    case 1:
        x <= a
        impossible, for all k, [a + tk, b + tk]
        does not include a.

    case 2:
        a < x < b
        if a < x, we can find N st a < (a + tk) < x
        so x in [a + tk, b + tk] forall k>N
        so x in liminf

    case 3:
        x = b
        we can find N st (a + tk) < b, and thus
        x in [a + tk, b + tk] forall k>N
        so x in liminf.

    case 4:
        x > b
        impossible, because we can find N st
        b < tk < x  st int_k>n [a + tk, b + tk]
        will not include x, forall k>N

    thus liminf En = (a, b]

    limsup En
    int_n U_k>n [a + tk, b + tk]

    consider U_k>n [a + tk, b + tk]
        we know (b + tn) > (b + tk) forall k>n
        thus (b + tn) is right end point of union
```

```
    for every e, st a < (a + e) < b
    (a + e) is included in the union
    because we can find N st a < (a + tk) < (a + e)
    forall k>N, thus (a + e) in [a + tk, b + tk]

    is (a) included?
    since tk > 0 forall k, a not in [a + tk, b + tk]
        for any k. thus not in union.
        same for points < a

    U_k>n [a + tk, b + tk] = (a, b + tn]
int_n (a, b + tn)
    once again,
    - this interval contains b because
        tn > 0 forall n, so
        (b + tn) > b forall n
    - it does not contain points above b, because
        for e > 0, we can find N st
        b < (b + tk) < b + e  forall k > N
        thus e not in (a, b+tk] forall k > N
(a,b]

thus limsup En = liminf En = (a,b] = lim En
```

```
d)
    E = (a,b)
    En = (a + tk, b + tk)

    liminf En
    U_n int_k>n (a + tk, b + tk)

    x = a
        impossible,
        since tk > 0, a not in any (a + tk, b + tk)

    a < x < b
        if a < x, we can find N st a < (a + tk) < x
        st x in (a + tk, b + tk) forall k>N
        x in liminf

    x = b
        we can find N st (a + tk) < b,
        then x in (a + tk, b + tk) forall k>N
        x in liminf
```

```
x > b
    impossible,
    we can find N st (b + tk) < x, and
    x not in (a + tk, b + tk) forall k>N

liminf En = (a,b]

limsup En
int_n U_k>n (a + tk, b + tk)

consider U_k>n (a + tk, b + tk)
    x = a
        impossible,
        a not in (a + tk, b + tk) for any k

    a < x < b
        we can find N st (a + tk) < x,
        and x in (a + tk, b + tk) forall k>N

    x >= b
        since tk decreasing, (b + tk) < (b + tn) forall k>n
        so (b + tn) is the right (open) end point of union

    U_k>n (a + tk, b + tk) = (a, b + tn)
int_n (a, b + tn)
    once again,
    - contains b because
        tn > 0 forall n
        b + tn > b forall n
        b in (a, b + tn) forall n
    - for e > 0, does not contain (b + e)
        we can find N st b < (b + tk) < (b + e),
        thus e not in (a, b + tk) forall k>N
(a, b]

thus limsup En = liminf En = (a,b] = lim En
```

e)

```
E = Q, rationals
(tn) rational forall but finitely many n

En = {rationals + tn}

liminf {rationals + tn}
```

```
        from (1.2) we know the liminf, limsup are the same
        as if we drop some finite number of terms, so let's
        drop any irrational numbers from (tn).

        since (tn) is a rational,
        {rationals + tn} = {rationals} forall n
            quik proof:
            {r + tn : r in Q} = {q : q in Q}
            ->
            x in LHS
            x = (r + tn) for r in Q
                since tn in Q, (r + tn) in Q
            x in Q
            x in RHS
            <-
            x in RHS
            x in Q
                add tn
            we have constructed (x + tn) : x in Q

        liminf {rationals} = {rationals}, a constant sequence

    limsup {rationals + tn} <- original tn
        = limsup {rationals + tn}
            tn only rational (1.2, drop irrationals)
        {rationals + tn} = {rationals} forall n
        = limsup {rationals}
            constant sequence
        = {rationals}

    thus liminf En = limsup En = Q = lim En

f)
    (tn) rational and irrational for infinitely many n

    liminf {rationals + tn}
    U_n int_k>n {rationals + tn}
        consider int_k>n {rationals + tn}
            if tn rational, {rationals + tn} = {rationals}
            if tn irrational, {rationals + tn} = {irrationals}
            we know both rational and irrational tn
            occur in k>n, so the intersection = nullset forall n
        U_n nullset
        nullset
```

```
    limsup {rationals + tn}
    int_n U_k>n {rationals + tn}
        consider U_k>n {rationals + tn}
        since both rational and irrational tn occur in k>n,
        thus union is R forall n
        int_n R
        R

    limsup En = R =/= nullset = liminf En, so lim En DNE

1.6
    I_A(x) = 1 if (x in A), 0 if (x in Ac)

    (An)
    A  subset  X

    a)
        lim An = A
        -> lim I_An = I_A

        lim An = A = limsup An = liminf An

        WTS: forall e > 0, there exists N, st forall k>N,
            | I_An(x) - I_A(x) | < e

        x in liminf An means
            we can find N0 st x in Ak forall k>N0
            thus I_Ak(x) = 1 forall k > N0

        since lim An exists, [lim An = liminf An]
            x in liminf and lim
            x in liminf
                x in Ak forall k>N0
                I_Ak(x) = 1 forall k > N0
            x in lim
                x in A
                I_A(x) = 1
            | I_A(x) - I_A(x) | = 0 < e forall k>N0

        thus we found a number, N0, satisfying WTS.

    b)
        lim I_An = I_A
        -> lim An = A
```

```
    forall e > 0, there exists N, st forall k>N,
        |I_Ak(x) - I_A(x)| < e

    WTS: limsup An = liminf An = A
        int_n U_k>n Ak = U_n int_k>n Ak = A

    pick e = 0.1, find N0 st forall k>N0,
        |I_Ak(x) - I_A(x)| < 0.1
        thus I_Ak(x) = I_A(x) forall k>N0
        so they are both (1 or 0) forall k>N0
        since I_A(x) is independent of k,
        they are both 1 forall k>N0, or both 0 forall k>N0

    if x in A, I_A(x) = 1
        I_A(x) = 1 = I_Ak(x) forall k>N0
        thus we found a number, N0, st x in Ak forall k>N0
        thus x in liminf An
        A  subset  liminf An
    if x in liminf An,
        there exists N1 st x in Ak forall k>N1
        thus I_Ak(x) = 1 forall k>N1
        thus lim I_Ak(x) = 1 = I_A(x)
        thus x in A
        liminf  subset  A
    A = liminf

    if x in limsup,
        x in An infinitely often.
        thus I_An(x) is 1 or 0 infinitely often.
        we know the limit I_An exists, so it is
            either 0 or 1.
        if the limit is 0, x is not in An i.o.
        so the limit must be 1
        that means there exists N2 st x in Ak forall k>N2
        but this means x in liminf
        limsup  subset  liminf
    always,
        liminf  subset  limsup
    liminf = limsup

    thus liminf = limsup = A

1.7
  AA sigma-algebra of X
```

```
Y  subset  X
BB = {A int Y : A in AA}

show BB sigma-algebra of Y

1)
    X in AA
    Y  subset  X
    X int Y = Y
    X int Y : X in AA
        is in BB
        Y in BB

2)
    B in BB
    WTS: Y \ B in BB
    Y \ (A int Y) for some A in AA
    Y int (A int Y)c
    Y int (Ac U Yc)
    (Y int Ac) U (Y int Yc)
    Y int Ac
        this is in BB since Ac in AA

3)
    Bn in BB
    WTS: U Bn in BB
    U Bn
    U (An int Y) for some An in AA
    (U An) int Y
        this is in BB since (U An) in AA

1.8
    AA collection of subsets of X
        - X in AA
        - A,B in AA => A \ B = A int Bc in AA

    WTS: AA is algebra
    1)
        X in AA
    2)
        A in AA.
        WTS: Ac in AA
        X,A in AA => X \ A = X int Ac = Ac in AA
    3)
        A,B in AA
```

WTS: A U B in AA
note (A U B)c = (Ac int Bc)
know Ac in AA by (2)
Ac \ B = (Ac int Bc) in AA
by (2), (Ac int Bc)c = (A U B) in AA

1.9

AA algebra of X
- for every increasing sequence (An) in AA, U An in AA.

WTS: AA is sigma-algebra.
1) X in AA
2) complements
3)
(Bn) in AA, WTS: U Bn in AA
define Cn = U_1..n Bk
Cn is increasing, U Cn in AA. but U Cn = U Bn in AA.

1.10

(X, AA) measurable space
(En) increasing sequence
U En = X

a)
AAn = {A int En : A in AA}

WTS: AAn sigma-algebra of En foreach n

by (1.7), AA is sigma-algebra of X,
En in AA => En subset X
then {A int En : A in AA} is a sigma-algebra, foreach n

b)
U AAn = AA ???
->
X in U AAn
X in AAn for some n
X = A int En for some n
A in AA
En in AA
X in AA
<-
B in AA
B either can or cannot be expressed as (A int En)
for some n,

```
        for some A in AA
    if it can,
        B = (A int En) for some n, for some A in AA
        B in AAn
        B in U AAn
    it it cannot, show that it cannot.
        En = (0, n/n+1)
        En increasing. its limit is (0,1)
        but X = (0,1) is not in any AAn, because
        U En = X
        (U En) int Ek = X int Ek
        Ek = (X int Ek)
        (0, k/k+1) = (0, k/k+1) =/= (0,1) for any k
    SO IT DOES NOT HOLD

1.11
  a)
    AAn increasing sequence of algebras of X,
    -> U AAn is algebra of X

    1)
        X in AAn, X in U AAn
    2)
        A in U AAn,
        A in AAn for some n,
        Ac in AAn
        Ac in U AAn
    3)
        A,B in U AAn
        A in AAn for some n
        B in AAm for some m
        A,B in AAl for l = max(n,m) since increasing
        (A U B) in AAl
        (A U B) in U AAn

  b)
    AAn decreasing sequence of algebras of X
    -> int AAn is algebra of X

    1)
        X in AAn foreach n
        X in int AAn
    2)
        A in int AAn
        A in AAn foreach n
```

```
            Ac in AAn foreach n
            Ac in int AAn
        3)
            A,B in int AAn
            A,B in AAn foreach n
            (A U B) in AAn foreach n
            (A U B) in int AAn

1.12
    (X, AA) msbl space
    E in AA is ATOM in the msbl space (X, AA) If
        - E =/= nullset
        - nullset, E are the only AA-msbl subsets of E

    E1, E2 distinct atoms in (X, AA)
    -> they are disjoint

    WTS: E1 int E2 = nullset
    E1 in AA
    E2 in AA
    (E1 int E2) in AA, since sigma-algebra
        (E1 int E2)  subset  E1
            E1 atom, so its subsets in AA must be nullset or E1

        case 1:
            (E1 int E2) = E1
            E1  subset  E2
            since E2 is an atom, E1 must be nullset or E2

            case 1.1:
                E1 = E2
                then they are not distinct atoms. contra.

            case 1.2:
                E1 = nullset
                E1 is an atom, and can't be nullset by def. contra.

            this case cannot happen.

        case 2:
            (E1 int E2) = nullset
            this is the only available option, so it must be true.

1.13
    CC arbitrary collection of subsets of X
```

```
a(CC) algebra generated by CC
sigma(CC)

sigma(CC) = int {AAa : a in A}
    intersection of all sigma-algebras containing CC
a(CC) = int {AAa : a in A}
    intersection of all algebras containing CC

a)
    a(a(CC)) = a(CC)
    a(CC) is smallest algebra containing CC.
    a(a(CC)) is smallest algebra containing a(CC)

    since a(CC) is algebra containing a(CC),
        a(CC) is included in the intersection on LHS
        a(a(CC))  subset  a(CC)
    by def, each algebra in intersection on LHS contains a(CC)
        a(CC)  subset  intersection
        a(CC)  subset  a(a(CC))

b)
    sigma(sigma(CC)) = sigma(CC)
    sigma(CC) is smallest sigma-algebra containing CC
    sigma(sigma(CC)) is smallest sigma-algebra containing sigma(CC)

    since sigma(CC) is a sigma-algebra containing sigma(CC),
        it is included in intersection on LHS
        sigma(sigma(CC))  subset  sigma(CC)
    by def, each sigma-algebra in intersection on LHS contains sigma(CC)
        sigma(CC)  subset  intersection
        sigma(CC)  subset  sigma(sigma(CC))

c)
    a(CC)  subset  sigma(CC)

    a sigma-algebra satisfies properties of algebra, so sigma(CC)
        is an algebra containing CC.
    thus it is included in the intersection on LHS.
    a(CC)  subset  sigma(CC)

d)
    CC finite collection, then a(CC) = sigma(CC)

    if CC is finite, any countable union of members of CC
        is a finite union. thus U Ak = U_1..n Ak in a(CC)
```

```
        meaning a(CC) contains all countable unions.

        thus a(CC) is a sigma-algebra containing CC,
            and it shows up in the intersection on RHS.
            sigma(CC)  subset  a(CC)
        sigma(CC) is an algebra containing CC,
            and shows up in intersection on LHS.
            a(CC)  subset  sigma(CC)

    e)
        sigma(a(CC)) = sigma(CC)

        sigma(a(CC)) is smallest sigma-algebra containing a(CC)
        from (c), know a(CC)  subset  sigma(CC)
        since sigma(CC) contains a(CC), it is included in the
            intersection on LHS
            sigma(a(CC))  subset  sigma(CC)
        CC  subset  a(CC)
            sigma(CC)  subset  sigma(a(CC))     [quik proof, p. 7]

BRUH MOMENT
    what is the limsup / liminf of (0, n/n+1) -> (0,1)
    x in liminf
        there exists n st x in Ak forall k>N
        x in (0,1)
        say x = 0.99999999999999999999999
        we can find n st 1 > n/n+1 > x > 0
        then x in Ak forall k>n
    x in limsup
        foreach n, there exists k>n st x in Ak
        x in (0,1)
        given n, given x,
            if x in An, return n
            if x not in An, find k like in liminf
    my question is from the sigma-algebra AAn question.
    we have a countable union of sigma-algebras, but it does
        not contain X for any n.
        the limit of the set sequence exists.
        but the sigma-algebras never contain X. wack.
    it seems the set limits contain things that
        are not included in Ak for any k.
    i can find N st x in Ak for any x in (0,1)
    but i cannot find N that works FORALL x in (0,1) simultaneously.
        because when i fix x, i am working with an interval
        (0,x] which can be covered by (0,n/n+1)
```

```
    but i cannot cover (0,1) with (0,n/n+1) for any n
once again, the limit seems to be describing the
    behavior of the set sequence rather than be a tangible
    set of points that are "always included" or something.

1.14
    (AAn) monotone sequence of sigma-algebras of X
    AA = lim AAn

    a)
        (AAn) decreasing,
        -> AA is sigma-algebra

        (AAn) monotone and decreasing, so limit is intersection
            AA = lim AAn = int AAn

        1)
            X in AAn foreach n
            X in int AAn
            x in AA

        2)
            A in int AAn
            A in AAn foreach n
            Ac in AAn foreach n
            Ac in int AAn

        3)
            Ak in int AAn
            Ak in AAn foreach n
            U Ak in AAn foreach n
            U Ak in int AAn

    b)
        (AAn) increasing
        -> AA is algebra
        -> AA not a sigma-algebra (construct example)

        AAn increasing, so limit is union
            AA = lim AAn = U AAn

        1)
            X in AAn foreach n
            X in U AAn
```

```
    2)
        A in U AAn
        A in AAn for some n
        Ac in AAn
        Ac in U AAn

    3)
        A,B in U AAn
        A in AAn for some n
        B in AAm for some m
        A,B in AAj for j = max(n,m) since increasing
        (A U B) in AAj
        (A U B) in U AAn

    AA not sigma-algebra:
    let CCn be a sequence of collection of subsets of X st:
        - X = [0,1] in CCn forall n
        - An = [0,1/n] in CCn
        - CCn  subset  CCn+1  forall n
    AAn are corresponding sigma-algebras generated by CCn
        CCn  subset  CCn+1
        sigma(CCn)  subset  sigma(CCn+1)
        AAn  subset  AAn+1
            sequence increasing
        consider int [0,1/n] = {0}
        {0} not in any AAn for any n
        thus {0} not in U AAn
        thus U AAn not closed under countable intersection

1.15
  CC = {A1..An} disjoint collection of nonempty subsets of X
  U_1..n Ai = X
  FF collection of all arbitrary unions of members of CC

  a)
    FF = sigma(CC)

    1)
        X = U_1..n Ai in FF
        this is a union of members of CC, so its in FF

    2)
        B in FF
        B = U Aj for {j}  subset  {1..n}
        X \ B = (U_1..n Ai) \ (U_j Aj : {j}  subset  {1..n})
```

```
        X \ B = U_k Ak : {k} = {j}c         [since disjoint]
            ok idk if this is rigorous enough
            say j is subset of {1..n}
            then X \ B is all members of {1..n} not in {j}
            so if {j} = {2,3}
            {j}c = {1..n} \ {2,3}
        this is a union of members of CC, so its in FF

    3)
        U Bn countable union of members of FF
            every member of FF is union of members of CC
        U Bn is union of members of CC
        U Bn in FF

    thus FF is sigma-algebra containing CC.
    sigma(CC)  subset  FF    [minimality]

    CC finite, so FF contains all finite unions of CC
    sigma(CC) is closed under finite unions
    FF  subset  sigma(CC)

  b)
    card(CC) = n

    FF contains all finite unions of CC.
        there are n sets, each can either be or not be included.
        total 2^n different unions

    card(FF) = 2^n = card(sigma(CC))

1.16
  CC = {Ai} disjoint collection of nonempty subsets of X
  U Ai = X
  FF collection of all arbitrary unions of members of CC

  a)
    1)
        X = U Ai is union of members of CC, thus in FF
    2)
        A in FF
        A is arbitrary union of members of CC
        A = U Ai : {i  subset  N}
        X \ A = (U Ai) \ (U Aj : {j  subset  N})
            since Ai disjoint,
            U Ak : {k} = {N \ {j}}
```

```
            this is a union of members of CC, thus in FF
    3)
        U An is countable union of members of FF
            each member of FF is union of members of CC
        U An is union of members of CC
        U An in FF

    FF is sigma-algebra, containing CC
    sigma(CC)  subset  FF    [minimality]

    now CC is countable, so unions of its members are
        at most countable unions.
        thus FF contains CC, and at most countable unions
            of members of CC
    FF  subset  sigma(CC)

b)
    the cardinality of CC is aleph-null
    each member of FF is a union of members of CC
        we can create a countable sequence of 0s and 1s
        where 0/1 at position n represents inclusion of An
        in the union.
    the cardinality of all such sequences is 2^card(CC)
        = 2^aleph-null

1.17
WTS: a sigma-algebra cannot be countably infinite,
    it ie either finite or uncountable.

consider CC an arbitrary collection of subsets of X.
if CC finite,
CC = {A1..An}
    since CC finite, for any point x in X,
    x belongs to at most n of the Ai.
        make a binary sequence from the Ai's that x belongs to.
        for example: x in A1, A3, A5
        the sequence is (10101)
        in binary this number is 21, so x in B_21
        let Bi be the set of all points with the same binary sequence.
            the intuition is that we partition the original space
            for every combination of the Ai's possible. so whenever
            some certain combination of Ai's intersect, that set
            will be one specific Bi. and the Bi's are disjoint.

            for example, B_21 will be all the points
```

```
                where A1 int A3 int A5

        if {Ai} has size n, {Bi} has size at most 2^n
        since {Bi} disjoint and finite, by (1.15) sigma({Bi}) is finite.
        sigma({Bi}) includes all of the original sets {A1..An}
            by taking unions of Bi.
        CC = {Ai}  subset  sigma({Bi})
        sigma(CC)  subset  sigma({Bi})

        consider the cardinality function for at most countable sets.
        1)
            card in [0,inf]
        2)
            card(nullset) = 0
        3)
            for An disjoint,
            card(U An) = sum card(An)
        thus "cardinality for at most countable sets" is a measure.

        by monotonicity, we can say
        card(sigma(CC)) <= card(sigma({Bi})) <= 2^(2^n)

        thus for CC finite collection of subsets,
            it has finite cardinality.

    if CC countable,
    CC = {Ai}
        consider the "best" scenario, where the Ai are disjoint.
        by (1.16) we know sigma(CC) has cardinality 2^aleph-null.
        for arbitrary CC, it can only be greater because set
        operations create even more sets.

    CC can only be finite, countable, or (more than countable).
    we showed the corresponding sigma-algebra has cardinality
    finite, 2^aleph-null, > 2^aleph-null
    thus there is no case where CC has sigma-algebra countably infinite.

1.18
    CC = {E1..En} finite collection of distinct,
        not necessarily disjoint subsets of X.

    DD = int (Ei or Eic)

    FF is collection of abitrary unions of DD
```

a)

any member of DD can be identified with a n-length sequence
representing if Ei is 1/0. there are 2^n sequences.

any 2 distinct members A,B of DD differ in at least 1 place in
this sequence, say at index i.
that means A is some set intersected with Ai
B is some set intersected with Aic
thus A,B necessarily disjoint.

b)

once again, consider all possible combinations. there are
n sets Ei. each Ei has either 1 or 0 corresponding to it.
thus we get 2 options per Ei. for a total of 2^n options.
there can be less, of course, if some sets coincide.

c)

[lol at this point i realized my set {Bi} from (1.17)
is the same thing as this. because if it's not in Ai,
it's in Aic... durr]
[so this question is basically what i did for (1.17)]

DD is a finite collection of disjoint sets.
FF is collection of arbitrary unions of members of DD.
by (1.15), FF = sigma(DD)
by (1.13 d), DD finite, a(DD) = sigma(DD)
thus FF = a(DD)

CC subset a(DD)
since Ei can be constructed from finite unions of
members of DD. for example E1 is union of all
sequences with 1 in first place
a(CC) subset a(DD)

DD subset a(CC)
since members of DD can be constructed by taking
finite intersections, and complements of Ei.
a(DD) subset a(CC)

a(CC) = a(DD) = FF

d)

cardinality of DD <= 2^n
by (1.15) sigma(DD) has cardinality <= 2^2^n
by (1.13 d) sigma(DD) = a(DD)

```
    by (c) a(CC) = a(DD)
    a(CC) has cardinality <= 2^2^n

e)
    sigma(DD) = a(CC)           [part (d)]
    sigma(DD) = sigma(a(CC))
    sigma(DD) = sigma(CC)       (1.13 e)
    a(CC) = sigma(CC)           [first line]

[thus we constructed a sigma-algebra for finite CC]
```

```
1.19
    CC arbitrary collection of subsets
    a(CC)

    for A in a(CC)
        there exists finite subcollection CC_A
        such that A in a(CC_A)

    CC_A  subset  CC
    a(CC_A)  subset  a(CC)
    A in both

    so basically A depends on a finite number of elements of CC

    a(CC) = U_A a(CC_A)

    argue by contradiction:
    assume A comes from a countable (or more) subset of CC.
        - if it includes X, that set's size doesn't change
        - if it takes complements, those sets' sizes don't change
        - if it's a finite union,
            then it needs to take a finite subset of the countable set.
            to use up the countable set, we need to keep taking
                finite subsets. we know that the result will still be a
                countable set of finite unions of the original countable set.
            of course i'm assuming they are all distinct.
            by repeatedly taking finite unions, we would never be able
                to reduce the countable set into a finite set.
            thus we would never be able to take a (finite) sequence of
                finite unions and get a member of a(CC).

        note: if we take a countable sequence of finite unions, that is
            equivalent to taking a countable union, and that is not
            guaranteed to be in a(CC).
```

```
we can use the same argument from the construction side:

    if we have a finite sequence of finite unions, it is still finite,
        and the result is in a(CC).

    if we have a countable sequence of finite unions, that is
        equivalent to a countable union,
        (because the index set would be countable)
        and not guaranteed to be in a(CC)

assuming my 2 arguments above are correct, we just showed that
    A in a(CC) must come from a finite subset of CC.
    and since A is constructed using the Algebra operations,
    A in a(CC_A)
```

```
1.20
    argue by contradiction:
    assume A depends on an uncountable subset of CC.
    - if A depends on X, 1 element, size doesn't change
    - if A is made by complements, size doesn't change
    - if A is made by countable unions,
        say each ingredient for A is made by taking out a
        countable number of items from the original uncountable set.
        even a countable number of countable unions is still
        countable, and will not be able to use up all the elements
        of the uncountable set.

    with a countable subcollection, we can easily construct an
        element A using countable unions, complements, and X.
```

```
1.21
    mu measure on sigma-algebra AA of X
    AA0 sub-sigma-algebra of AA,
        AA0 is sigma-algebra of X
        AA0  subset  AA

    call mu restricted to AA0 pu

    1)
        mu(A) in [0,inf] for A in AA
        mu(A) in [0,inf] for A in AA0  subset  AA
        thus pu in [0,inf]

    2)
```

```
        mu(nullset) = 0
        X in AA, X in AA0 => nullset in AA0
        thus pu(nullset) = 0

    3)
        En disjoint in AA0 =>
            En disjoint in AA
        U En in AA0 [by sigma-algebra] =>
            U En in AA
        (En), (U En) both in AA and AA0 thus
            pu(En) = mu(En) foreach n
            pu(U En) = mu(U En)
        pu(U En) = mu(U En) = sum mu(En) = sum pu(En)

1.22
    (X, AA, mu)
    E1, E2 in AA
        mu(E1 U E2) + mu(E1 int E2) = mu(E1) + mu(E2)

    additivity;
    (E1 U E2)
        = (E1 \ E2) U (E1 int E2) U (E2 \ E1) disjoint
    E1 = (E1 \ E2) U (E1 int E2)
        = (E1 int E2c) U (E1 int E2) disjoint
    E2 = (E2 \ E1) U (E1 int E2)
        = (E2 int E1c) U (E2 int E1) disjoint

    mu(E1 U E2)
    mu(E1 int E2c) + mu(E1 int E2) + mu(E2 int E1c)

    RHS:
    mu(E1) + mu(E2)
    mu(E1 int E2c) + mu(E1 int E2) + mu(E2 int E1c) + mu(E2 int E1)
    mu(E1 U E2) + mu(E2 int E1)

1.23
    (X,AA)
    muk measure on AA
    ak >= 0 foreach k

    mu = sum_k ak muk

    1)
        muk in [0,inf]
        ak muk in ak * [0,inf] = [0,inf] since ak >= 0
```

```
        sum_k ak muk in [0,inf] since countable sum of
            numbers in [0,inf]
            quik proof:
            sum_k inf = inf
            for each k, sum_1..k inf = inf
            thus lim_k sum_1..k inf = inf
            thus sum_k inf = inf

            [ok i went and checked Tao's book for the def
            of a series (lol) and i think this is true:]

            the partial sum Sk = sum_1..k = inf for each k
            thus Sk diverges
            lim Sk = inf
            thus lim_n sum_1..n inf = inf
            thus sum_n inf = inf

    2)
        sum_k ak muk(nullset)
        sum_k ak 0
        sum_k 0
        Sk = sum_1..k 0 = 0 foreach k

        for any e > 0, we can find N st Sk < e forall k>N
            (mainly N = 1)
        thus lim_n Sn = 0
        thus lim_n sum_1..n 0 = 0
        thus sum_n 0 = 0

    3)
        En in AA
        mu(U En)
            WTS: = sum_n mu(En)
                 = sum_n sum_k ak muk(En)
        sum_k ak muk(U En)
        sum_k ak sum_n muk(En)
        sum_k sum_n ak muk(En)
        sum_n sum_k ak muk(En)  [interchange sums bc they are non-neg]
        sum_n mu(En)

1.24
    X = (0,inf)
    JJ = {Jk : k in N}
        Jk = (k-1,k]   for k in N
```

```
AA is collection of all arbitrary unions of members of JJ

for A in AA, define mu(A)
    as number of elements of JJ in A

a)
    [i literally fucking showed this by applying (1.16) and then
    proceeded to re-prove this. i am literally disabled]

    (by 1.16, Its the sigma-algebra)
    quik proof:
    U (k-1,k] = (0,inf)
    ->
    x in (k-1,k] for some k in N
    then x in (0,k+1)  subset  (0,inf)
    <-
    x in (0,inf)
    then 0 < x < inf
    since x < inf, x is finite
    since x finite, there exists k in N st (k-1) < x <= k
    thus x in (k-1,k] for some k in N
    thus x in U (k-1,k]


    1)
        X = U (k-1,k] is arb union of members of JJ, in AA
    2)
        B in AA, so B = U_a Aa is arb union of members of JJ
            since JJ is countable, union is at most countable
        Bc
        X \ B
        (U (k-1,k]) \ (U_a (k-1,k])
            LHS is union over k in N
            RHS is union over subset of N
        since the sets (k-1,k] disjoint, this is equal to
            U (k-1,k] over {N \ {a}}
            (is this rigorous enough?)
        this is once again an at-most-countable union of
            members of JJ, so it is in AA.
    3)
        (Bn) in AA
        Bn = U (k-1,k]
            each Bn is at-most-countable union of members of JJ
        U Bn
            is a countable union of at-most-countable unions of
```

```
            members of JJ
        thus it is countable.
        a countable union of members of JJ is in AA.

b)
    1)
        mu is defined to be the number of elements of JJ in A.
            JJ is countable.
        it can be 0 for an empty union.
        it is a positive number for a finite union.
        it is infinity for a countable union.
        thus mu in [0,inf]
    2)
        mu(nullset)
        there are 0 members of JJ in here.
        = 0
    3)
        (Ak) disjoint sequence in AA

        each Ak is arbitrary (at most countable) union of
            members of JJ
        thus (U Ak) is (at most countable) union of members of JJ

        case 1:
            at least one Ak has infinite cardinality
            mu(Ak) = inf <= mu(U Ak)  [monotonicity]
                mu(U Ak) = inf        [mu in [0,inf]]
            mu(Ak) = inf <= sum mu(Ak)  [mu non-neg]
                inf <= sum mu(Ak) <= inf
                    showed above that countable sum inf = inf
                thus sum mu(Ak) = inf
            mu(U Ak) = sum mu(Ak)

        case 2:
            countable union of Ak with finite cardinality
            base case:
            mu(U_1..1 Ak) = mu(Ak)
            mu(U_1..2 Ak) = mu(A1) + mu(A2)  [disjoint, finite]
            ind step:
            mu(U_1..n Ak)
            mu(An U U_1..n-1 Ak)
            mu(An) + mu(U_1..n-1 Ak)
            sum_1..n mu(Ak)
            thus foreach n,
                mu(U_1..n Ak) = sum_1..n mu(Ak)
```

```
            case 2.1:
                sum_1..n mu(Ak) converges to L

        case 3:
            finite union of Ak with finite cardinality, the rest 0s
            by induction, the thing has finite size and
            we can say mu(U Ak) = sum mu(Ak)

        [i wasted 3 hours on this, but the answer is
        "it follows directly" meaning that the number of
        members in the disjoint union is the sum of the
        individual parts]

    c)
        (An) = (n, inf)

        lim An
        liminf An = limsup An
        U_n int_k>n (k,inf) = int_n U_k>n (k,inf)
        ->
        given n, int_k>n (k,inf) is empty
            because if x in it, x finite,
            there exists N st N-1 < x < N,
            then (N+1,inf) does not contain x
        U_n nullset
        nullset
        <-
        given n,
            U_k>n (k,inf) = (n,inf)
        int_n (n,inf)
        nullset  [as explained above]

        mu(lim An) = mu(nullset) = 0

        lim_n mu(An)
        mu(An) = mu(U_k>n (k-1,k]) = inf  [countable union of disjoint sets]
        lim_n inf = inf

        thus lim_n mu(An) = inf =/= 0 = mu(lim An)

1.25
    (X,AA,mu) sigma-finite
        exists (En) st U En = X
        mu(En) < inf
```

```
consider the cumulative union of En
    U_1..n Ek
    increasing set sequence

now define Fn as the "new content" of this cumulative union
F1 = U_1..1 Ek = E1
Fn = (U_1..n Ek) \ (U_1..n-1 Ek)
    = En \ (U_1..n-1 Ek)
    = En int (U_1..n-1 Ek)c
    = En int (int_1..n-1 Ekc)
    Fn disjoint by construction
    U Fn = U cumulative_union by construction
    U Fn = U En = X
    mu(Fn) <= mu(En) < inf foreach n
```

```
1.26
    BBr Borel sigma-algebra of R
    Lebesgue mul
        measure on BBr
        interval I in R, mul(I) = len(I)

    a)
        (En) st
            lim En exists
            lim mul(En) DNE

        according to (thm 1.28 d), if
            lim En exists
            En finite foreach n
            -> lim mu(En) exists and the limits are equal

        to get contrapositive, we must have En not finite

        [ok im thinking... if En is infinite foreach n,
        then the measure is infinity, and so its limit
        is just infinity? apparently infinity is not
        considered a limit... meaning i did a bunch of
        stuff wrong. lol.]

        En = R
        lim En = R
        lim mul(En) = inf <- apparently this means limit DNE

    b)
```

```
    (En) st
        lim mul(En) exists
        lim En DNE

    consider En = (0, (n mod 2) + 1) = (0,1),(0,2),(0,1),...
    mul(En) = 1
    lim mul(En) = 1

    lim En
    U_n int_k>n En = int_n U_k>n En
    ->
    U_n (0,1)
    (0,1)
    <-
    int_n (0,2)
    (0,2)

c)
    (En) st
        lim En exists
        lim mul(En) exists
        mul(lim En) =/= lim mul(En)

    once again to get contrapositive of (thm 1.28 d) we must
        have En not finite

    consider En = (n,inf)
    lim En
        liminf En = limsup En
        U_n int_k>n (n,inf) = int_n U_k>n (n,inf)
        ->
        consider int_k>n
            this is nullset because for any x,
                (ceil(x)+1, inf) wont include it
        U_n nullset = nullset
        <-
        consider U_k>n (n,inf) = (n,inf)
        int_n (n,inf) = nullset
        lim En = nullset
    mul(En) = inf - n = inf foreach n
    thus lim mul(En) = inf =/= 0 = mul(lim En)

    [aaaand i realized i just repeated prob 1.24 d]

d)
```

```
    x in R
    En = (x - 1/2n, x + 1/2n)
    len(En) = 1/n = mul(En)

    lim En
        liminf En = limsup En
        U_n int_k>n (x-1/2n, x+1/2n) = int_n U_k>n (x-1/2n, x+1/2n)
        ->
        consider int_k>n
            only {x} in intersection
        U_n {x}
        {x}
        <-
        consider U_k>n
            = (x-1/2n, x+1/2n)
        int_n (x-1/2n, x+1/2n)
        {x}  [same reasoning as above]

    thus {x} is a countable intersection of open sets
    {x} in BBr

    lim_n mul(x-1/2n, x+1/2n)
    lim_n 1/n = 0

    by theorem 1.28,
        lim En exists = {x}
        En finite measure (1/n)
        -> then lim mu(En) exists = 0
        -> and mu(lim En) = lim mu(En)
    thus mu({x}) = lim mu(En) = 0

e)
    we know Q is a collection of individual points.
        Q = { p/q : p in Z, q in N }
            let's say P in N, then we can just union 0,+p,-p
        Q = {0} U
            { p/q : p in N, q in N } U
            {-p/q : p in N, q in N }
            <just to make it a bit easier>

    for p in N,
        for q in N,
            { p/q } in BBr, by part (d)
        the countable union over q in N is in BBr.
        this set is { {p/1}, {p/2}, {p/3}, ...}
```

```
        call this set Sp, dependent on p
    the countable union over p in N is in BBr.
    this is all positive rationals.
    make a copy for negatives.
    add {0}
    and we get Q.

    mul({p/q}) = 0 foreach p, foreach q
    mul(Sp) = 0 since countable sum of 0
    mul(U Sp) = sum mul(Sp) = 0 since countable sum of 0
    mul(negative copy) = 0
    mul({0}) = 0

f)
    irrationals = {Q}c in BBr

    by lemma (1.25) part (3):
        rationals, R, in BBr
        rationals  subset  R
        mul(rationals) = 0 < inf
        -> mul(R \ rationals) = mul(R) - mul(rationals)
        -> mul(irrationals) = inf - 0 = inf

g)
    U_{{x} : 0 < x < 1}
    = (0,1)




1.27
    (R, Power(R))
    mu(E) = number of elements in E, if E finite
          = inf, if E infinite

    a)
        1)
            number of elements in a set can be
                0, finite, or infinity
            thus mu in [0,inf]
        2)
            mu(nullset)
            there are 0 elements in nullset
            = 0
        3)
            En disjoint sequence
```

```
        mu(U En)

        case 1:
            mu(U En) infinite

            case 1.1:
                at least one En infinite
                then sum mu(En) infinite, and they are equal

            case 1.2:
                all En finite
                then since En disjoint,
                there is a countable set of En with positive measure
                then sum mu(En) will be infinite as well

        case 2:
            mu(U En) finite
            that means the countable union must be finite
            finite union of finite sets. we can use induction
            to show mu(U En) = sum mu(En)

    b)
        we must show there is no sequence (En) st
            U En = R
            mu(En) < inf foreach n

        [i think the only way to prove these things in math is to
        assume they exist]

        assume (En) exists, satisfying
            U En = R
            mu(En) < inf foreach n
        this means En is a finite set
        the countable union of finite sets is at most countable
        but R is uncountable

1.28
    (X,AA,mu) finite
    CC = {El} disjoint collection of members of AA st
        mu(El) > 0

    for a countable union of Els,
        mu(U El) = sum mu(El)  since disjoint
        < inf since finite measure space
```

```
    if CC is uncountable,
    we can take out a countable subset.
    this subset will have mu(U En) = sum mu(En) > 0

    OK THEYRE DISJOINT, POZ MEASURE, AND COUNTABLY MANY



    [i think the answer is that an uncountable collection
    of disjoint sets with measure >0 cannot be finite.

    we know measure is additive for disjoint countable union
    a countable sum can be finite.

    the answer has to be that an uncountable sum of positive
    numbers cannot be finite, but idk how to prove that]

    if CC countable,
    mu(U_l El) = sum_l mu(El), union/sum over l countable
        < mu(X) < inf
    if CC not countable, for every countable sequence, we
        can still find more El in CC.

    by (theorem 1.28 d)
        define Fn = U_1..n El be some countable subseq
        its increasing, limit exists
        its finite
        -> lim mu(Fn) exists = mu(lim Fn)

1.29
    X countably infinite
    AA = Power(X)
    mu(E) = 0 if E finite
          = inf if E infinite

    a)
        additive:
            E1, E2 disjoint in AA

            f/f
                mu = 0

            f/inf
                mu = 0 + inf = inf

            inf/inf
```

```
            mu = inf + inf = inf

        not countably additive:
            En countable disjoint set
            mu(U En) = inf  since U En is countable set

            sum mu(En)
                if at least 1 En is inf, this = inf
                if all En finite, = 0 and thus not countably additive

    b)
        show X limit of inc seq En with mu(En) = 0 forall n, mu(X) = inf

        X is countably infinite so mu(X) = inf

        since X countable, enumerate its elements as {an : n in N}
            thus we have X = U {an}

        define En = U_1..n {ak}
            increasing set seq

        WTS: lim En = U {an} = X
        liminf En = limsup En
        U_n int_k>n Ek = int_n U_k>n Ek
        ->
        int_k>n U_1..k {aj} = U_1..n {ak}
        U_n U_1..n {ak} = U_n {an}
        <-
        U_k>n U_1..k {aj} = U_k {ak}
        int_n U_k {ak} = U_k {ak}

        [also could've just said it's increasing so lim = union...]

        mu(En) = 0 since it's a finite union of elements. foreach n.

1.30
    X arbitrary infinite set.
    subset A of X is COFINITE if Ac is finite.
    AA collection of all finite and cofinite subsets of X.

    a)
        1)
            Xc = nullset is finite
            thus X is cofinite
            thus X in AA
```

```
2)
    A in AA
    A is finite or cofinite
    if A finite,
        Ac is cofinite because Acc = A finite.
        thus Ac in AA.
    if A cofinite,
        Ac is finite and in AA.
3)
    A,B in AA
    finite + finite
        A,B both finite
        (A U B) finite

    finite + cofinite
        A finite
        B cofinite
        Bc finite
        (Ac int Bc)  subset  Bc finite
        (Ac int Bc)c cofinite
        (A U B) cofinite

    cofinite + cofinite
        A cofinite
        B cofinite
        Ac finite
        Bc finite
        (Ac U Bc) finite
        (Ac U Bc)c cofinite
        (A int B) cofinite

        (A int Bc)  subset  Bc finite
        (A \ B) finite

        (Ac int B)  subset  Ac finite
        (B \ A) finite

        (A \ B) U (B \ A) = (A triangle B) finite
        (A triangle B) U (A int B) in AA
            because we already proved finite + cofinite
        = (A U B) cofinite

b)
    countable union not in it
    probably countable union of finite sets is not finite,
```

```
    but complement is...

    X = [0,1]
    An = {1\n} finite foreach n
    (U An) = U {1/n} not finite
    (U An)c = (int Anc)
        includes at least (1/2,1) which is not finite

1.31
  X arbitrary uncountable
  subset A is co-countable if Ac is countable
  AA collection of all countable and cocountable subsets.

  a)
    Xc = nullset countable, so X in AA
  b)
    if A in AA, A is countable or cocountable
    if A countable,
        Ac is cocountable since Acc = A is countable
    if A is cocountable,
        Ac is countable and in AA
  c)
    An in AA
    if at least one An is countable,
        int An is countable, in AA
    if all An cocountable,
        all Anc are countable
        (int An) = (U Anc)c
        countable union of countable sets is countable
        its complement in AA

  [what they're saying is that if A,Ac are both
  uncountable, then they cannot be created by the union op.
  uncountable sets can only be got through complements.
  for example:
  X = [0,1]
  AA countable/cocountable subsets
  A = [1,0.5) not in AA
  Ac = [0.5,1] not in AA
  then we cant arrive at these sets thru complements,
  and neither through union/int of individual points.
  [assume there are no intervals in AA]]

  [but in Borel[0,1] we have uncountable union of points
  happens to be an interval in the sigma-algebra. but we
```

worked backwards: we got points out of intervals rather
than intervals as uncountable unions]

1.32

X infinite
AA algebra of finite and cofinite
mu(A) = 0 if A finite, 1 if cofinite

a)

mu additive on AA
A1, A2 disjoint in AA

finite + finite
A1, A2 finite -> (A1 U A2) finite, in AA
mu(A1) + mu(A2) = mu(A U B) = 0

finite + cofinite
A finite, B cofinite
we showed in 1.30 that
(A U B) cofinite
0 + 1 = 1

cofinite + cofinite
we showed that (A int B) cofinite
but here A,B disjoint
inpossible case

[i remember this soln from rosenthal lol]

b)

X countably infinite
mu not countably additive

countable additivity is:
for En disjoint
* * * we know at most 1 En is cofinite
mu(U En) = sum mu(En)

to be in AA, we must have (U En) be finite or cofinite
if (U En) finite, then finite additiity holds
if (U En)c is finite, (U En) is cofinite, and
at most 1 En is cofinite
mu(U En) = 1

```
        but we can create a set En such that
            sum mu(En) = 0
        for example, since X countable,
            enumerate it as xi. Then the
            set sequence {xi} is disjoint and
            sum mu(xi) = 0

c)
    X countably infinite
    X is limit of increasing seq An
        mu(An) = 0 foreach n,
        mu(X) = 1

    ok as in the previous part, enumerate X as xi
    let En = {xn} this is disjoint seq
    let Fn = U_1..n Ek be cumulative union
        it has measure 0 by construction

    we know the limit of Fn is
        lim Fn = U Fn = U En = U {xi}
        this countable union of {xi} is exactly X

d)
    X uncountable
    -> mu countably additive on algebra AA

    to be countably additive:
        for En disjoint,
        mu(U En) = sum mu(En)

    to be in AA, (U En) finite or cofinite
        if (U En) finite, finite additivity holds
        if (U En)c finite, (U En) cofinite
            at most 1 En cofinite
            but also at least 1 En cofinite, because
                if we had each En be finite, (U En) is
                countably infinite, complement is uncountable.
                    (last time X was countable
                    and (U En)c = finite set of points)
                    thus (U En) not in AA.

        knowing (U En) cofinite,
            contains exactly 1 cofinite En,
            mu(U En) = 1
            sum mu(En) = 1
```

```
1.33
    X uncountable
    AA sigma-algebra of countable & cocountable subsets
    mu(A) = 0 if A countable, 1 if A cocountable

    show mu countably additive
        for An disjoint
        mu(U An) = sum mu(An)

    0 An cocountable
        all Ans countable
        (U An) countable
        mu(U An) = sum mu(An) = 0

    1 An cocountable
        say A1 cocountable
        (U_2.. An) countable
        A1c countable
        (int Anc)  subset  A1c  countable
        (int Anc) countable
        (U An)c countable
        (U An) cocountable
        mu(U An) = 1

        sum mu(An) = 1
            since only 1 of the An is cocountable

    >1 An cocountable
        A1, A2 cocountable
        A1c countable
        A2c countable
        (A1c U A2c) countable
        (A1c U A2c)c cocountable
        (A1 int A2) cocountable
            since their complement is countable,
            must have uncountable cardinality since X uncountable
        but hypothesis is that A1, A2 disjoint. so this case cant happen

1.34
    (X,AA,mu)
    collection {Al}  subset  AA
        almost disjoint if l1 =/= l2
        -> mu(Al1 int Al2) = 0
```

```
a) {An} almost disjoint.


    induction.

    base case:
    mu(A1 U A2) = mu(A1 A2c) + mu(A2 A1c) + mu(A1 A2)
    mu(A1) + mu(A2)
        mu(A1 A2) + mu(A1 A2c) + mu(A2 A1) + mu(A2 A1c)
        mu(A1 U A2) + mu(A1 A2)
        mu(A1 U A2)    since mu(A1 A2) = 0

    assume:
    sum_1..n mu(Ak) = mu(U_1..n Ak)

    know
    mu(A1) + mu(A2) = mu(A1 U A2) + mu (A1 A2)
        replace A1 with An+1
        replace A2 with U_1..n Ak
        mu(An+1) + mu(U_1..n Ak) = mu(U_1..n+1 Ak) + mu(An+1 int U_1..n Ak)

        mu(An+1 int U_1..n Ak)
        mu(U_1..n [Ak An+1]) <= sum mu(Ak An+1)  [subadditivity]
        = 0

    thus foreach n, mu(U_1..n Ak) = sum_1..n mu(Ak)
    ->
    foreach n, mu(U_1..n Ak) <= sum mu(An)   [since mu non-neg]
    <-
    sum_1..n mu(Ak) = mu(U_1..n Ak) <= mu(U An)  [monotonicity]

    [ok i think this is a general method: if result holds for each n,
    foreach n sum is less than mu(U) by monotonicity,
    and foreach n, finite-union less than sum since mu non-neg]

b)
    (An) satisfies
        mu(U An) = sum mu(An)
        mu(An) < inf
        -> An almost disjoint
        -> WTS: mu(An int Am) = 0

    by (lemma 1.25)
        (U_1..n Ak)  subset (U Ak)
        mu(U_1..n Ak) < inf  [since each one < inf]
```

```
-> mu((U Ak) \ (U_1..n Ak)) = mu(U Ak) - mu(U_1..n Ak)
-> mu(U_n+1.. Ak) = mu(U Ak) - mu(U_1..n Ak)

starting with n = 1, we get
mu(U_2.. Ak) = sum_2.. mu(Ak)

know:
mu(A1) + mu(A2) = mu(A1 U A2) + mu(A1 A2)
    replace A1 with A1
    replace A2 with (U_2.. Ak)

mu(A1) + mu(U_2.. Ak) = mu(A1 U (U_2.. Ak)) + mu(A1 int (U_2.. Ak))
mu(A1) + sum_2.. mu(Ak) = mu(U Ak) + mu(A1 int (U_2.. Ak))
sum mu(Ak) = mu(U Ak) + mu(A1 int (U_2.. Ak))
-> mu(A1 int (U_2.. Ak)) = 0
        since (A1 int Ak)  subset  (A1 int (U_2.. Ak))  [for k >= 2]
        mu(A1 int Ak) = 0   [monotonicity]  [for k >= 2]

        [TODO: can we even do this? sum mu(Ak) might be infinity
        but we have eqn x = x + e, so maybe e=0 still?]


induction.
induction hypothesis:
assume mu(An int Ak) = 0 for n=1..n
this implies mu(U_1..n+1 Ak) = sum_1..n+1 mu(Ak)

mu(U_n+2.. Ak) = sum_n+2.. mu(Ak)
    [this result requires mu(U_1..n Ak) = sum_1..n mu(Ak)]

know
mu(A1) + mu(A2) = mu(A1 U A2) + mu(A1 A2)
    replace A1 with (U_1..n+1 Ak)
    replace A2 with (U_n+2.. Ak)
mu(U_1..n+1 Ak) + mu(U_n+2.. Ak) = mu((U_1..n+1 Ak) U (U_n+2.. Ak)) + mu((U_1..n+1 Ak) int (U_n+2..
    know mu(U_1..n+1 Ak) = sum_1..n+1 mu(Ak) because already
        proved that int with An is 0 for n = 1..n
sum mu(Ak) = sum mu(Ak) + mu((U_1..n+1 Ak) int (U_n+2.. Ak))

    [TODO: once again, sum mu(Ak) might be infinity?]

    mu((U_1..n+1 Ak) int (U_n+2.. Ak)) = 0
    mu(An+1 int (U_n+2.. Ak)) = 0            [monotonicity]
    mu(An+1 int Ak) = 0   [for k >= n+2]     [monotonicity]
```

```
        n = 2
        mu(U_3.. Ak) = sum_3.. mu(Ak)

        mu(A1) + mu(A2) = mu(A1 U A2) + mu(A1 A2)
            replace A1 with (A1 U A2)
            replace A2 with (U_3.. Ak)
        mu(A1 U A2) + mu(U_3.. Ak) = mu((A1 U A2) U (U_3.. Ak)) + mu((A1 U A2) int (U_3.. Ak))
        mu(A1 U A2) + sum_3.. mu(Ak) = mu(U Ak) + mu((A1 U A2) int (U_3.. Ak))
            know mu(A1 U A2) = mu(A1) + mu(A2) since proved n=1
        sum mu(Ak) = sum mu(Ak) + mu((A1 U A2) int (U_3.. Ak))
        mu((A1 U A2) int (U_3.. Ak)) = 0
        mu [A1 int (U_3.. Ak)] U [A2 int (U_3.. Ak)] = 0
            consider just right part. by monotonicity,
            [A2 int (U_3.. Ak)]  subset  [A1 int (U_3.. Ak)] U
                                            [A2 int (U_3.. Ak)]
            thus
            mu[A2 int (U_3.. Ak)] = 0
                by monotonicity,
                (A2 int Ak)  subset  (A2 int (U_3.. Ak))  [for k >=  3]
                mu(A2 int Ak) = 0


    C)
        we can't use (lemma 1.25)
        if mu(A1) = inf,
            then the condition
                mu(U An) = sum mu(An)
                is satisfied trivially,
                and the rest of the sequence can be arbitrary.

1.35
    (X,AA,mu)

    a)
        A = B  iff  A tri B = nullset
        ->
        A tri B
        A tri A
        (A \ A) U (A \ A)
        nullset U nullset
        <-
        (A \ B) U (B \ A) = nullset
```

```
(A int Bc) U (B int Ac) = nullset
(A int Bc)  subset  <the union> = nullset
(B int Ac)  subset  <the union> = nullset
show A  subset  B
    assume x in A but not in B.
    then x in (A int Bc)
    but this is a nullset.
    thus if x in A, x must be in B.
show B  subset  A
    assume x in B but not in A.
    then x in (B int Ac)
    but this is a nullset.
    thus if x in B, x must be in A.
```

b)

```
A U B  =  (A int B) U (A tri B)
->
x in A U B
x in A, B, or both
x in (A \ B), (B \ A), (A int B)
x in (A tri B) U (A int B)
<-
x in RHS
x in (A int B) or x in (A tri B)
x in (A int B) or x in (A \ B) or x in (B \ A)
x in (A int B) U (A \ B) U (B \ A)
x in A and B, or x in just A, or x in just B
x in A or B
x in (A U B)
```

c)

```
(A tri B)  subset  (A tri C) U (C tri B)
(A \ B) U (B \ A)  subset  (A \ C) U (C \ A) U (C \ B) U (B \ C)
(A int Bc) U (B int Ac)  subset
    (A int Cc) U (C int Ac) U (C int Bc) U (B int Cc)
    [(C int Ac) U (C int Bc)] U [(A int Cc) U (B int Cc)]
    [C int (Ac U Bc)] U [Cc int (A U B)]

    x can either be in C or Cc.
    if x in Cc,
        then it is definitely also in (A U B).

    if x in C,
        if x in (A \ B), x in Bc.
        if x in (B \ A), x in Ac.
```

```
            in both cases, x in (Ac U Bc).

    d)
        (A tri B)  subset  (A tri C) U (C tri B)       [part c]
        mu(A tri B) <= mu[(A tri C) U (C tri B)]       [monotonicity]
            <= mu(A tri C) + mu(C tri B)               [subadditivity]

    e)
        A U B  =  (A int B) U (A tri B)                [part b]
        mu(A U B)  =  mu((A int B) U (A tri B))
            need to show (A int B) and (A tri B) are disjoint
            (A int B) int [(A int Bc) U (B int Ac)]
            [((A int B) int (A int Bc)) U
                ((A int B) int (B int Ac))]
            nullset U nullset
            thus disjoint
            = mu(A int B) + mu(A tri B)                [additivity]

    f)
        mu(A tri B) = 0
        -> mu(A) = mu(B)

        (A U B) = (A \ B) U (B \ A) U (A int B)
        (A U B) = (A tri B) U (A int B)
        mu(A U B) = mu(A tri B) + mu(A int B)    [additivity]
        mu(A U B) = mu(A int B)                  [mu(A tri B) = 0]

        (A int B)  subset  A  subset  (A U B)
        (A int B)  subset  B  subset  (A U B)

        mu(A int B) <= mu(A) <= mu(A U B)
        mu(A U B) = mu(A int B) <= mu(B) <= mu(A U B) = mu(A int B)

        thus mu(A) = mu(B) = mu(A U B) = mu(A int B)

1.36
    (X,AA,mu) finite
    p(A,B) = mu(A tri B)
        p(A,B) in [0, mu(X)]
        p(A,B) = p(B,A)
        p(A,B) <= p(A,C) + p(C,B)
        p not a metric bc p(A,B) doesnt imply A = B

    relation ~ for AA
    A ~ B means
```

```
    mu(A tri B) = 0

a)
    1)
        A ~ A means
        mu(A tri A)
        mu((A \ A) U (A \ A))
        mu(nullset U nullset)
        0
    2)
        A ~ B
        mu(A tri B) = 0
        mu((A \ B) U (B \ A)) = 0
        mu((B \ A) U (A \ B)) = 0
        mu(B tri A) = 0
        B ~ A
    3)
        A ~ B, B ~ C
        mu(A tri B) = 0 = mu(B tri C)

        from (1.35 d)
        mu(S1 tri S2) <= mu(S1 tri S3) + mu(S3 tri S2)
            replace S1 with A
            replace S2 with C
            replace S3 with B

        mu(A tri C) <= mu(A tri B) + mu(B tri C) = 0

b)
    we partition on equivalence classes of sets
        with the same measure since
        mu(A tri B) = 0
        -> mu(A) = mu(B)  [prob 1.35 f]

    p* takes 2 equivalence classes. it picks a representative
        from each one, and returns their symmetric difference.

    A' in [A]
    B' in [B]
        mu(A' tri A) = 0
            mu(A) = mu(A')
        mu(B' tri B) = 0
            mu(B) = mu(B')
    mu(A' tri B')
```

by (1.35 d)
mu(A' tri B') <= mu(A' tri A) + mu(A tri B')
mu(A tri B') <= mu(A tri B) + mu(B tri B')

mu(A tri B) <= mu(A tri A') + mu(A' tri B)
mu(A' tri B) <= mu(A' tri B') + mu(B' tri B)

mu(A tri B) <= mu(A' tri B')
mu(A' tri B') <= mu(A tri B)
thus they are equal.

c)

imagine [A] as a pile of things with A as representative
[B] is a pile of things with B as representative
p* takes 2 piles and pick representatives form both,
and puts them in mu()

1)

p*([A],[B]) = mu(A tri B) in [0,mu(X)]

2)

p*([A],[B]) = 0 <=> [A] = [B]
->
p*([A],[B]) = 0
-> mu(A tri B) = 0
-> A ~ B
-> A,B belong to the same equivalence class
-> [A] = [B]
<-
[A] = [B]
p*([A],[B])
p*([A],[A])
mu(A tri A) = 0

3)

p*([A],[B])
mu(A tri B)
mu(B tri A)
p*([B],[A])

4)

WTS: p*([A],[B]) <= p*([A],[C]) + p*([C],[B])

p*([A],[B])
mu(A tri B) <= mu(A tri C') + mu(C' tri B) [1.35 d]

```
                for any C' in AA,
                C' belongs to equivalence class [C]
                [C] in [AA]
                mu(A tri C') = mu(A tri C)
                mu(C' tri B) = mu(C tri B)
            = mu(A tri C) + mu(C tri B)
            = p*([A],[C]) + p*([C],[B])

            [basically pass from class to representative,
            take measure of representative]

        [TODO: never used the fact that its finite???]




NOT SURE:
1.28
1.34b
```

```
% -----------------------------------------------------------------
    TODO: whats the result where if an <= a for all n, then
        lim an <= a ???

    TODO schilling proof that f inverse is good for set ops
    TODO various associoation and distribution properties of A \ B
% -----------------------------------------------------------------

https://math.stackexchange.com/questions/172167/intuitive-interpretation-of-limsup-and-liminf-of-sequences-of-se

https://math.stackexchange.com/questions/107931/lim-sup-and-lim-inf-of-sequence-of-sets

https://math.stackexchange.com/questions/485815/intuition-behind-the-definition-of-a-measurable-set

https://mathoverflow.net/questions/34007/demystifying-the-caratheodory-approach-to-measurability

https://jmanton.wordpress.com/2017/08/24/intuition-behind-caratheodorys-criterion-think-sharp-knife-and-shrink-w

Questions to ask a human:
- we say x occurs either finite or infinitely often. how do we know that
    there isn't a single occurence, an infintie number of times away?
```

maybe because sequences are made of components for each n, and the limits are defined from suprema/infima as lower/upper bounds. meaning we don't care what happens an infinite time away, we only care about describing its behavior at each n. and if something is true for each n, then that describes the limit.

$$\liminf_{n\to\infty} A_n = \bigcup_{n=1}^{\infty}\bigcap_{k=n}^{\infty} A_k$$

$$\limsup_{n\to\infty} A_n = \bigcap_{n=1}^{\infty}\bigcup_{k=n}^{\infty} A_k$$