Discrete Mathematics
Homework 4

Ethan Bolker

December 5, 2014

Exercises

1. Is forming the cartesian product of two sets a commutative operation?

   Let S and T be sets. Investigate when/whether

   $$S \times T = T \times S.$$

   Start this problem by working a few small examples! When you understand what's going on, write your answer as a theorem like this:

   Theorem 1. *$S \times T = T \times S$ if and only if [something] or [something else].*

   Proof. Suppose $S \times T = T \times S$. Then ... so something or something else is true.

   Conversely, suppose neither something nor something else is true. Then ... so $S \times T \neq T \times S$. □

   Solution

   Recall that $S \times T$ is itself a set, whose elements consist of all the ordered pairs (a, b) where $a \in S$ and $b \in T$.

   I tried a small example. (Some of you chose much bigger small examples, so did more work but didn't learn any more.)

   Let $S = \{1, 2\}$ and let $T = \{1, 2, 3\}$. Then

   $$S \times T = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3)\}$$

   and

   $$T \times S = \{(1, 1), (1, 2), (2, 1), (2, 2), (3, 1), (3, 2)\}.$$

   These sets are not equal since $(2, 3)$ is in the first one but not the second. In a set, the elements have no particular order, but in an *ordered pair* they do: $(2, 3) \neq (3, 2)$.

   That example suggests that I will be able to prove that $S \times T \neq T \times S$ when $S \neq T$.

   We've seen that $\emptyset$ has interesting and sometimes non-intuitive properties, so I should investigate $S \times \emptyset$. Aha! That cartesian product is empty! (We noted that in class.)

   Now I know enough information to state and prove the theorem.

   Theorem 1. *$S \times T = T \times S$ if and only if 1) $S = T$, or 2) if S or $T = \emptyset$.*

   Proof. "If" is easy. It's essentially what I did when figuring out the theorem. If (2) holds, say $T = \emptyset$. Then

   $$S \times \emptyset = \emptyset \times S = \emptyset$$

   whatever S happens to be, empty or not.

If (1) holds (so $S = T$) then obviously

$$S \times T = S \times S = T \times S.$$

Now I have to prove "only if" – the converse. Suppose $S \times T = T \times S$. If either S or T is empty then I've proved conclusion (2), so I only have to deal with the case when neither is empty. To show that S and T are the same set I need to show that every element of S is in T and vice versa. Well, suppose $s \in S$. Since $T \neq \emptyset$, pick any $t \in T$. Then $(s, t) \in S \times T = T \times S$. That says $s \in T$. That means $S \subseteq T$. This argument is perfectly symmetrical in S and T, so $T \subseteq S$. Thus $S = T$ and I've proved "only if". □

2. The algebra of symmetric differences.

We'll use $A \oplus B$ for the symmetric difference of subsets A and B of a universe U.

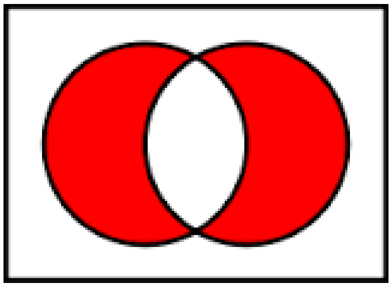
- Show that $\oplus$ is associative.

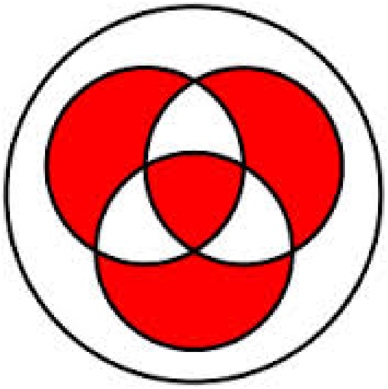
Solution

I need to show that

$$(A \oplus B) \oplus C = A \oplus (B \oplus C).$$

There are two good ways to do this. The first is to find a *symmetric* way to describe what elements are in either side. If you work at it you can see that x is such an element just when it's in exactly one of A, B and C or it's in all three. Here are the Venn diagrams from `http://en.wikipedia.org/wiki/Exclusive_or` for $A \oplus B$ and for $(A \oplus B) \oplus C$:





The second good way is to remember that we can think of subsets as bitstrings. When we do that, the symmetric difference corresponds to adding mod 2, with no carrying. (We did that in class.) But addition mod 2 is associative!

A third way to do the problem is with truth tables. Those of you who tried it succeeded.

A fourth way is to work out both sides of the expression using the definition of $\oplus$ in terms of set union, intersection and difference. That's tedious but satisfactory. It's hard to read to check for typographical errors.

- Find a subset $I \subseteq U$ such that for every A we have $A \oplus I = A$.

Solution

Just take I to be the empty set. That corresponds to the bitstring that's all zeroes. Adding it to anything makes no change.

You can also check that $A \oplus \emptyset = A$ directly from the definition.

- For each A, find a B such that $A \oplus B = I$.

Solution

In the previous part of the problem I showed that $I = \emptyset$. For this part, let $B = A$! That works because $A \oplus A$ is just the empty set.

- True or false:

$$\text{If } A \oplus B = A \oplus C \text{ then } B = C.$$

Just saying "true" (or "false") isn't an answer. You must explain why.

Solution

That's true. To see why, just see what happens if you add ($\oplus$) A to both sides and use the results of the previous parts of the problem:

If

$$A \oplus B = A \oplus C$$

then

$$\begin{aligned} B &= \emptyset \oplus B \\ &= (A \oplus A) \oplus B \\ &= A \oplus (A \oplus B) \\ &= A \oplus (A \oplus C) \\ &= (A \oplus A) \oplus C \\ &= \emptyset \oplus C \\ &= C. \end{aligned}$$

Hints:

- If you have taken or are taking abstract algebra lots of this should look familiar.

  Solution This set with this operation is a group. In fact it's the direct product of many copies of the two element group.

- Thinking about how $\oplus$ works with the bit vector representation of subsets might save you a lot of thinking and writing.

  Solution I did that above.

3. A binary relation on a set may be reflexive, symmetric or transitive. Call those properties R, S, and T for short. There are eight possible truth value combinations for those properties. For each of the eight, find an example. (So, for example, one of your answers should exhibit a binary relation that satisfies R and $\sim S$ and $\sim T$.)

   Please find elegant examples – the best are either everyday situations where the properties are obviously what you claim they are, or binary relations built on the smallest possible finite set that will do the job. You can use answers to previous problems when they work here.

   Solution

   There are lots of correct answers. Here are a few.

   - RST: Any equivalence relation: equals, is parallel to, is the same color as, ...
   - $RS \sim T$: Shares a common language with. Shares a parent.
   - $R \sim ST$: Is greater than or equal to.
   - $R \sim S \sim T$: $\{(1,1),(1,2),(2,2),(2,3),(3,3)\}$.
   - $\sim RST$: The *empty* relation on a nonempty set!
   - $\sim RS \sim T$: Is not equal to. $\{(1,2),(2,1)\}$.
   - $\sim R \sim ST$: Is greater than. Is an ancestor of. Is a proper subset of.
   - $\sim R \sim S \sim T$: $\{(1,2),(2,3)\}$.

4. Let $\star$ be the relation on $\mathbb{Z} \times \mathbb{Z}$ defined by

$$(a,b) \star (c,d) \iff ad = bc.$$

   (a) Describe all the pairs (a,b) such that $(a,b) \star (4,6)$.

   Solution

   $(a,b) \star (4,6)$ just when $6a = 4b$. That's the same as $a = \frac{2}{3}b$. It's *almost* the same as $\frac{a}{b} = \frac{2}{3}$ – but you have to worry about $a = b = 0$.

   Solution

   (b) Show that $\star$ is not an equivalence relation, but is an equivalence relation if you consider only the ordered pairs in which the second coordinate is not 0.

   $\star$ is reflexive:

$$(a,a) \star (a,a) \text{ because } a \times a = a \times a.$$

$\star$ is symmetric:

$$\begin{aligned}(a,b)\star(c,d) &\implies ad=bc\\ &\implies cb=da\\ &\implies (c,d)\star(a,b).\end{aligned}$$

$\star$ is not transitive, since $(1,1)\star(0,0)$ and $(0,0)\star(2,1)$ but $(1,1)\not\star(2,1)$.

But if we look only at the ordered pairs whose second coordinate is not 0 then $\star$ is transitive:

If none of b, d or e is 0 then

$$\begin{aligned}(a,b)\star(c,d) &\implies ad=bc \implies \frac{a}{b}=\frac{c}{d}\\ (c,d)\star(e,f) &\implies cf=de \implies \frac{c}{d}=\frac{e}{f}\end{aligned}$$

so $\frac{a}{b}=\frac{e}{f}$ and thus $(a,b)\star(e,f)$.

(c) Prove: given (a,b) with $b\neq 0$ there is a unique pair (m,n) equivalent to (a,b) for which $\gcd(m,n)=1$. (Hint, if needed: think about the previous part of the problem.)

Solution

Use the hint. Among all the fractions equal to a/b there is one that's in lowest terms.

(d) Show that there is a bijection between $\mathbb{Q}$ (the rational numbers) and the partition $\mathcal{Q}$ corresponding to $\star$.

Solution

I need to build a bijection $f:\mathbb{Q}\to\mathcal{Q}$.

If $q\in\mathbb{Q}$ is not 0 then write $q=m/n$ in lowest terms and let $f(q)$ be the block of $\mathcal{Q}$ that contains (m,n). Let $f(0)$ be the block of $\mathcal{Q}$ that contains $(0,1)$. It's not hard to show that f is a bijection, but it's tedious. I won't bother. If you got this far and decided not to bother either you got full credit.

5. For each of the following sets, provide an argument showing that it's countable, or that it's not.

Solution

Note: No one got this problem all correct. Very few people even managed to make a good start on it.

(a) The set of all partitions of $\mathbb{N}$.

I know that the set $\mathcal{P}(\mathbb{N})$ of all subsets of $\mathbb{N}$ is uncountable. Partitions of $\mathbb{N}$ are even more complicated than subsets of $\mathbb{N}$ so they are probably uncountable too. In fact they are, and the answer to the next part of the problem proves it.

(b) The set of all partitions of $\mathbb{N}$ into a finite number of blocks.

I will show that the set of partitions of $\mathbb{N}$ into just 2 blocks is uncountable. The set of partitions into two blocks is a (proper) subset of the set of partitions into finitely many blocks, so that set will be uncountable too.

To show that the set $\mathcal{S}$ of partitions into 2 blocks is uncountable I will find a bijection f between $\mathcal{S}$ and the set $\mathcal{T}$ of subsets of $\mathbb{N}$ that contain the number 1. There are clearly uncountable many of those.

Suppose $P\in\mathcal{S}$. Then $P=\{A,B\}$ where A and B are disjoint and cover $\mathbb{N}$. That means 1 is in just one of A and B. Define $F(P)$ to be the block that does not have 1 in it.

If you think carefully about this definition of F you will see that it matches the two-block partitions of $\mathbb{N}$ with the subsets of $\mathbb{N}$ that don't contain 1, so it's a bijection.

(c) The set of all partitions of $\mathbb{N}$ for which each block is finite.

This set is uncountable too. To show that I will find a bijection from the the set $\mathcal{I}$ of infinite subsets of $\mathbb{N}$ to a subset of the set $\mathcal{F}$ of partitions of $\mathbb{N}$ for which all the blocks are finite. Since there are uncountably many infinite subsets of $\mathbb{N}$ there are uncountably many of those partitions.

The best way to illustrate my bijection is with an example. Suppose I start with the infinite set

$$\{2, 3, 5, 7, 11, \ldots\}.$$

(That happens to be the set of prime numbers.)

I will match that with the partition

$$\{\{1, 2\}, \{3\}, \{4, 5\}, \{6, 7\}, \{8, 9, 10, 11\}, \ldots\}$$

That is, I use the numbers in the infinite set as the end numbers in the blocks of the partition. This way I construct a partition with infinitely many finite blocks. It's a very special kind of partition – the numbers in each block are consecutive. Since I have shown there are uncountably many of this special kind of partition, there are uncountably many when I think about all the partitions.

6. Counting computer programs

   Each of these questions can be answered in a sentence or two if you really understand the work we did in class on counting infinities.

   Solution

   Note: No one got this problem all correct either. Very few people even managed to make a good start on it.

   (a) Show that there are only countably many finite strings of (ascii) characters.

   There are 128 ascii characters, or 256 if you count 8 bits. The exact number is irrelevant. I'm sorry I even used "ascii" in the question since it led people down a useless track.

   If you are writing finite strings with an alphabet with c characters there are exactly c^n of them of length n. What really matters is that there are only finitely many of each length. That means the set of all finite strings is a countable union of finite sets, so it's countably infinite.

   (b) Show that in any particular programming language there are only countably many computer programs that accept an integer as input and produce a boolean value as output.

   This question has nothing to do with integer input and boolean output. I wish I hadn't said that here. All that matters is counting how many computer programs are possible.

   A computer program in any particular language is just a finite string of characters. It doesn't matter what the program does, or whether it's correct. It's just a finite string of characters – think about it as a text file. So there are only countably many possible programs.

   (c) Show that there are uncountably many functions from $\mathbb{Z}$ to $\{T, F\}$.

   We saw in class that the functions from $\mathbb{Z}$ to a two element set correspond exactly to the subsets of $\mathbb{Z}$. There are uncountably many subsets, so uncountably many functions.

   (d) Show that for any particular computer language there are functions from $\mathbb{Z}$ to $\{T, F\}$ that can't be implemented by a program in that language.

   I just showed that there are only countably many programs while there are uncountably many functions. So there aren't enough programs to go around. There must be functions that you can't compute with a program.

   This has nothing to do with fact that in any particular language the integer data type might have only finitely many values. Even though Java `Integer` handles only values less than `Integer.MAXINT` you can easily write Java programs that accept arbitrarily big integers as input strings. You model them internally using the `BigInteger` class.

   (e) If you have a language and a function that can't be computed using that language you can create a new and better language to compute it. If you do that over and over again will you have a language that computes all functions?

   No. This is the same flaw in reasoning that defeats the attempt to defeat Cantor diagonalization by adding the "set that was left out" to the list. After you've done that, some other set is still left out.

7. (Optional, but I'd love to see it.)

   Read the fictional treatment of Hilbert's Hotel at `http://www.c3.lanl.gov/mega-math/workbk/infinity/inhotel.html`.

   Then rewrite the ending so that the narrator's scheme for housing the people on the infinitely many infinite busses succeeds (as it will). Then have the fire caused in some interesting way by the failure (discussed in class) of any scheme that claims to accomodate all the committees of guests.

   Try to copy the author's style.

```
% Math 320 hw4
%
\documentclass{article}
\pagestyle{empty}
\usepackage[textheight=10in]{geometry}

\usepackage{mathtools, nccmath}

\usepackage{amssymb}
\usepackage{amsthm}
\usepackage{listings}
\usepackage{graphicx}
\usepackage{verbatim}

% hyperref should be (nearly) the last package loaded
\usepackage{hyperref}

\usepackage{xparse}

%This very cool macro comes from
%http://tex.stackexchange.com/questions/209863/how-to-add-mathematical-notation-of-a-set
%
% \set{ stuff ; something }
% expands to
% { stuff | something }
%
\DeclarePairedDelimiterX{\set}[1]{\{}{\}}{\setargs{#1}}
\NewDocumentCommand{\setargs}{>{\SplitArgument{1}{;}}m}
{\setargsaux#1}
\NewDocumentCommand{\setargsaux}{mm}
{\IfNoValueTF{#2}{#1} {#1\,\delimsize|\,\mathopen{}#2}}%{#1\:;\:#2}
\newtheorem{theorem}{Theorem}

\newcommand{\coursehome}
{http://www.cs.umb.edu/~eb/320}

\title{Discrete Mathematics \\
Homework 4
}
\author{Ethan Bolker}
%\date{September 1, 2014}

\newcommand{\ZZ}{\mathbb{Z}}
\newcommand{\NN}{\mathbb{N}}
\newcommand{\QQ}{\mathbb{Q}}
\newcommand{\RR}{\mathbb{R}}


%create (mod n) macro
\newcommand{\mm}[1]{%
\ensuremath{(\text{mod } #1)}}

\begin{document}


\maketitle

\noindent
%Due: ???
%
```

```
%This homework covers problems in an area loosely called ‘‘sets and
%functions’’. You can read about it in B\&W at
%\url{http://cseweb.ucsd.edu/~gill/BWLectSite/Resources/C1U4SF.pdf}.
%
%I will add to this homework from time to time.
%
%See the source code for some cool \verb+\set+ macros to construct
%these sets in \LaTeX:
%
%\[
%D = \set{x \in \NN ; 1\leq x\leq 100}
%\]
%
%The delimiters adjust to the size of the contents in the * version:
%\[
%E = \set*{x \in \QQ ; -\frac{1}{2}\leq x \leq \frac{1}{2}}
%\]
%
%You also can have a manual adjustment with an optional argument to \verb+\set+:
%\[
%E = \set[\big]{x \in \QQ ; -\mfrac{1}{2}\leq x \leq \mfrac{1}{2}}
%\]
%
%And you can define sets as simple lists:
%\[\text{Unit fractions}= \set*{\mfrac{1}{1}, \mfrac{1}{2},\mfrac{1}{3}, \dots }
%\]
%
%
\section*{Exercises}

\begin{enumerate}

\item Is forming the cartesian product of two sets a commutative
  operation?

Let $S$ and $T$ be sets. Investigate when/whether
%
\begin{equation*}
S \times T = T \times S .
\end{equation*}

Start this problem by working a few small examples! When you
understand what’s going on, write your answer as a theorem like this:

\begin{theorem}
$S \times T = T \times S$  if and only if [something] or [something else].
\end{theorem}
\begin{proof}
Suppose $S \times T = T \times S$. Then \ldots so something or
something else is true.

Conversely, suppose neither something nor something else is true. Then
\ldots so
$S \times T \not = T \times S$.
\end{proof}

\textbf{Solution}

Recall that $S \times T$ is itself a set, whose elements consist of
all the ordered pairs $(a,b)$ where $a\in S$ and $b \in T$.
```

```
I tried a small example. (Some of you chose much bigger small examples,
so did more work but didn’t learn any more.)

Let $S=\set{1,2} $ and let $T=\set{1,2,3}$. Then

\begin{equation*}
S \times T = \set{(1,1),(1,2),(1,3),(2,1),(2,2),(2,3)}
\end{equation*}

and

\begin{equation*}
T \times S = \set{(1,1),(1,2),(2,1),(2,2),(3,1),(3,2)}.
\end{equation*}

These sets are not equal since $(2,3)$ is in the first one but not the
second. In a set, the elements have no particular order, but in an
\emph{ordered pair} they do:
$(2,3)\ne (3,2).$

That example suggests that I will be able to prove that $S \times T
\ne T \times S$ when $S \ne T$.

We’ve seen that $\emptyset $ has interesting and sometimes
non-intuitive properties, so I should investigate
$S \times \emptyset$. Aha! That cartesian product is empty! (We noted
that in class.)

Now I know  enough information to state and prove the theorem.

\addtocounter{theorem}{-1}
\begin{theorem}
$S \times T = T \times S$ if and only if 1) $S = T$, or 2) if $S$ or
  $T = \emptyset.$
\end{theorem}
\begin{proof}

‘‘If’’ is easy. It’s essentially what I did when figuring out the theorem.
If (2) holds, say $T = \emptyset$. Then
\begin{equation*}
S \times \emptyset = \emptyset \times S = \emptyset
\end{equation*}
%
whatever $S$ happens to be, empty or not.

If (1) holds (so $S = T$) then obviously
%
\begin{equation*}
S \times T = S \times S = T \times S .
\end{equation*}

Now I have to prove ‘‘only if’’ -- the converse.
Suppose $S \times T = T \times S$. If either $S$ or $T$ is empty then
I’ve proved conclusion (2), so I only have to deal with the case when
neither is empty. To show that $S$ and $T$ are the same set  I need to
show that every element of $S$ is in $T$ and vice versa. Well, suppose
$s \in S$. Since $T \ne \emptyset$, pick any $t \in T$. Then
$(s,t) \in S \times T = T \times S$. That says $s \in T$. That means
$S \subseteq T$.  This argument is perfectly symmetrical in $S$ and
```

```
$T$, so $T \subseteq S$. Thus $S=T$ and I’ve proved ‘‘only if’’.
\end{proof}

\item The algebra of symmetric differences.

We’ll use $A \oplus B$ for the symmetric difference of subsets $A$ and
$B$ of a universe $U$.

\begin{itemize}
\item Show that $\oplus$ is associative.

\textbf{Solution}

I need to show that
\begin{equation*}
(A \oplus B) \oplus C = A \oplus (B \oplus C) .
\end{equation*}

There are two good ways to do this. The first is to find a
\emph{symmetric} way to describe what elements are in either side. If
you work at it you can see that $x$ is such an element just when it’s
in exactly  one of $A$, $B$ and $C$ or it’s in all three. Here are the
Venn diagrams from \url{http://en.wikipedia.org/wiki/Exclusive_or}
for $A \oplus B$ and for $(A \oplus B) \oplus C$:

\begin{center}
\includegraphics[width=2in]{abvenn}
\quad
\includegraphics[width=2in]{abcvenn}
\end{center}

The second good way is to remember that we can think of subsets as
bitstrings. When we do that, the symmetric difference corresponds to
adding mod 2, with no carrying. (We did that in class.) But addition
mod 2 is associative!

A third way to do the problem is with truth tables. Those of you who
tried it succeeded.

A fourth way is to work out both sides of the expression using the
definition of $\oplus$ in terms of set union, intersection and
difference. That’s tedious but satisfactory. It’s hard to read to
check for typographical errors.

\item Find a subset $I \subseteq U$ such that for every $A$ we have $A
  \oplus I = A$.

\textbf{Solution}

Just take $I$ to be the empty set. That corresponds to the bitstring
that’s all zeroes. Adding it to anything makes no change.

You can also check that $A \oplus \emptyset = A$ directly from the
definition.

\item For each $A$, find a $B$ such that $A \oplus B = I$.

\textbf{Solution}

In the previous part of the problem I showed that $I = \emptyset$. For
```

```
this part, let $B = A$! That works because $A \oplus A$ is just the
empty set.

\item True or false:
%
\begin{equation*}
\text{If } A \oplus B = A \oplus C \text{ then } B = C.
\end{equation*}

Just saying ‘‘true’’ (or ‘‘false’’) isn’t an answer. You must explain
why.

\textbf{Solution}

That’s true. To see why, just see what happens if you add ($\oplus$)
$A$ to both sides and use the results of the previous parts of the
problem:

If
\begin{equation*}
A \oplus B = A \oplus C
\end{equation*}
%
then
%
\begin{align*}
B & = \emptyset \oplus B\\
  & = (A \oplus A) \oplus B \\
  & = A \oplus ( A \oplus B ) \\
  & = A \oplus ( A \oplus C )\\
  & = (A \oplus A) \oplus C \\
  & = \emptyset \oplus C\\
  & = C .
\end{align*}

\end{itemize}

Hints:
\begin{itemize}
\item If you have taken or are taking abstract algebra lots of this
  should look familiar.

\textbf{Solution} This set with this operation is a group. In fact
it’s the direct product of many copies of the two element group.

\item Thinking about how $\oplus$ works with the bit vector
  representation of subsets might save you a lot of thinking and
  writing.

\textbf{Solution} I did that above.
\end{itemize}

\item  A binary relation on a set may be reflexive, symmetric or
transitive. Call those properties $R$, $S$, and $T$ for short. There are
eight possible truth value combinations for those properties. For each
of the eight, find an example. (So, for example, one of your answers
should exhibit a binary relation that satisfies $R$ and $\sim S$ and
$\sim T$.)

Please find elegant examples -- the best are either everyday situations
```

```
where the properties are obviously what you claim they are, or
binary relations built on the smallest possible finite set that will
do the job. You can use answers to previous problems when they work here.

\textbf{Solution}

There are lots of correct answers. Here are a few.

\begin{itemize}
\item $RST$: Any equivalence relation: equals, is parallel to, is the
  same color as, \ldots
\item $RS \sim T$: Shares a common language with. Shares a parent.
\item $R \sim S T$: Is greater than or equal to.
\item $R \sim S \sim T$: $\set{(1,1), (1,2), (2,2), (2,3), (3,3)}$.
\item $\sim RST$: The \emph{empty} relation on a nonempty set!
\item $\sim RS \sim T$: Is not equal to. $\set{(1,2), (2,1)}$.
\item $\sim R \sim S T$: Is greater than. Is an ancestor
  of. Is a proper subset of.
\item $\sim R \sim S\sim T$: $\set{(1,2),(2,3)}$.
\end{itemize}

\item Let $\star$ be the relation on $\ZZ \times \ZZ$ defined by
%
\begin{equation*}
(a,b) \star (c,d) \iff ad = bc .
\end{equation*}

\begin{enumerate}

\item Describe all the pairs $(a,b)$ such that $(a,b) \star (4,6)$.

\textbf{Solution}

$(a,b) \star (4,6)$ just when $6a = 4b$. That’s the same as
$a = \frac{2}{3}b$. It’s \emph{almost} the same as
$\frac{a}{b} = \frac{2}{3}$ -- but you have to worry about $a=b=0$.

\textbf{Solution}

\item Show that $\star$ is not an equivalence relation, but is an
  equivalence relation if you consider only the ordered pairs in which
  the second coordinate is not $0$.

$\star$ is reflexive:
%
\begin{equation*}
(a,a) \star (a, a) \text{ because } a \times a = a \times a .
\end{equation*}

$\star$ is symmetric:
%
\begin{align*}
(a,b) \star (c,d) & \implies ad = bc \\
                  & \implies cb = da \\
                  & \implies (c,d) \star (a,b) .
\end{align*}
$\star$ is not transitive, since $(1,1) \star (0,0)$ and
$(0,0) \star (2,1)$ but $(1,1) \not\star (2,1)$.
```

```
But if we look only at the ordered pairs whose second coordinate is
not $0$ then $\star$ is transitive:

If none of $b, d$ or $e$ is $0$ then
%
\begin{align*}
(a,b) \star (c,d) & \implies ad = bc \implies \frac{a}{b} = \frac{c}{d} \\
(c,d) \star (e,f) & \implies cf = de \implies \frac{c}{d} = \frac{e}{f}
\end{align*}
%
so $\frac{a}{b} = \frac{e}{f}$ and thus $(a,b) \star (e,f)$.

\item Prove: given $(a,b)$ with $b \neq 0$ there is a unique pair $(m,n)$
  equivalent to $(a,b)$ for which $\gcd(m,n) = 1$. (Hint, if needed:
  think about the previous part of the problem.)

\textbf{Solution}

Use the hint. Among all the fractions equal to $a/b$ there is one
that’s in lowest terms.

\item Show that there is a bijection between $\QQ$ (the rational
  numbers) and the partition $\mathcal{Q}$  corresponding to $\star$.

\textbf{Solution}

I need to build a bijection $f : \QQ \to \mathcal{Q}$.

If $q \in \QQ$ is not $0$ then write $q = m/n$ in lowest terms and
let $f(q)$ be the block of $\mathcal{Q}$ that contains $(m,n)$. Let
$f(0)$ be the block of $\mathcal{Q}$ that contains $(0,1)$. It’s not
hard to show that $f$ is a bijection, but it’s tedious. I won’t
bother. If you got this far and decided not to bother either you got
full credit.

\end{enumerate}

\item For each of the following sets, provide an argument showing that
  it’s countable, or that it’s not.

\textbf{Solution}

\emph{Note: No one got this problem all correct. Very few people even
managed to make a good start on it.}

\begin{enumerate}
\item The set of all partitions of $\NN$.

I know that the set $\mathcal{P}(\NN)$ of all subsets of $\NN$ is
uncountable. Partitions of $\NN$ are even more complicated than
subsets of $\NN$ so they are probably uncountable too. In fact they
are, and the answer to the next part of the problem proves it.

\item The set of all partitions of $\NN$ into a finite number of
  blocks.

I will show that the set of partitions of
$\NN$ into just $2$ blocks is uncountable.
The set of partitions into two blocks is a (proper) subset of the set
of partitions into finitely many blocks, so that set will be
```

```
uncountable too.

To show that the set $\mathcal{S}$ of partitions into $2$ blocks is
uncountable I will find a bijection $f$ between $\mathcal{S}$ and the set
$\mathcal{T}$ of subsets of $\NN$ that contain the number $1$. There
are clearly uncountable many of those.

Suppose $P \in \mathcal{S}$. Then $P = \set{A,B}$ where $A$ and $B$ are
disjoint and cover $\NN$. That means $1$ is in just one of $A$ and
$B$. Define $F(P)$ to be the block that does not have $1$ in it.

If you think carefully about this definition of $F$ you will see that
it matches the two-block partitions of $\NN$ with the subsets of $\NN$
that don't contain $1$, so it's a bijection.

\item The set of all partitions of $\NN$ for which each block is finite.

This set is uncountable too. To show that I will find a bijection from
the the set $\mathcal{I}$ of infinite subsets of $\NN$ to a subset of
the set $\mathcal{F}$ of partitions of $\NN$ for which all the blocks are
finite. Since there are uncountably many infinite subsets of $\NN$
there are uncountably many of those partitions.

The best way to illustrate my bijection is with an example. Suppose I
start with the infinite set
%
\begin{equation*}
\set{2, 3, 5, 7, 11,  \ldots}.
\end{equation*}
%
(That happens to be the set of prime numbers.)

I will match that with the partition
\begin{equation*}
\left\{ \set{1,2}, \set{3}, \set{4,5},
\set{6,7}, \set{8,9,10,11}, \ldots \right\}
\end{equation*}

That is, I use the numbers in the infinite set as the end numbers in
the blocks of the partition. This way I construct a partition with
infinitely many finite blocks. It's a very special kind of partition
-- the numbers in each block are consecutive. Since I have shown there
are uncountably many of this special kind of partition, there are
uncountably many when I think about all the partitions.

\end{enumerate}

\item Counting computer programs


Each of these questions can be answered in a sentence or two if you
really understand the work we did in class on counting infinities.

\textbf{Solution}

\emph{Note: No one got this problem all correct either. Very few
  people even  managed to make a good start on it.}


\begin{enumerate}
```

```
\item Show that there are only countably many finite strings of (ascii)
  characters.

There are 128 ascii characters, or 256 if you count 8 bits. The exact
number is irrelevant. I’m sorry I even used ‘‘ascii’’ in the question
since it led people down a useless track.

If you are writing finite strings with an alphabet with $c$ characters
there are exactly $c^n$ of them of length $n$. What really matters is
that there are only finitely many of each length. That means the set
of all finite strings is a countable union of finite sets, so it’s
countably infinite.

\item Show that in any particular programming language there are only
  countably many computer programs that accept an integer as input and
  produce a boolean value as output.

This question has nothing to do with integer input and
boolean output. I wish I hadn’t said that here. All that matters is
counting how many computer programs are possible.

A computer program in any particular language is just a finite string of
characters. It doesn’t matter what the program does, or whether it’s
correct. It’s just a finite string of characters -- think about it as
a text file. So there are only countably many possible programs.

\item Show that there are uncountably many functions from $\ZZ$ to
  $\set{T,F}$.

We saw in class that the functions from $\ZZ$ to a two element set
correspond exactly to the subsets of $\ZZ$. There are uncountably many
subsets, so uncountably many functions.

\item Show that for any particular computer language there are
  functions from $\ZZ$ to   $\set{T,F}$ that can’t be implemented by a
  program in that language.

I just showed that there are only countably many programs while there
are uncountably many functions. So there aren’t enough programs to go
around. There must be functions that you can’t compute with a program.

This has nothing to do with fact that in any particular language the
integer data type might have only finitely many values. Even though
Java \verb!Integer! handles only values less than
\verb!Integer.MAXINT! you can easily write Java programs that accept
arbitrarily big integers as input strings. You model them internally
using the \verb!BigInteger! class.

\item If you have a language and a function that can’t be computed
  using that language you can create a new and better language to
  compute it. If you do that over and over again will you have a
  language that computes all functions?

No. This is the same flaw in reasoning that defeats the attempt to
defeat Cantor diagonalization by adding the ‘‘set that was left out’’
to the list. After you’ve done that, some other set is still left out.
\end{enumerate}

\item (Optional, but I’d love to see it.)
```

```
Read the fictional treatment of Hilbert's Hotel at
\url{http://www.c3.lanl.gov/mega-math/workbk/infinity/inhotel.html}.

Then rewrite the ending so that the narrator's scheme for housing the
people on the infinitely many infinite busses succeeds (as it will). Then
have the fire caused in some interesting way by the failure (discussed
in class) of any scheme that claims to accomodate all the committees
of guests.

Try to copy the author's style.
\end{enumerate}
\newpage

\verbatiminput{\jobname}

\end{document}
```