A Manifesto for a Free Society

Andrew Yu

Revision 0.1

Revision 0.1.

This document is still a draft, and will probably stay that way in a few decades. Please apply your critical thinking skills when reading. We cannot guarentee that what we think of as facts are all true, and we cannot guarentee that other ideas apply to you. Generally, you should take this approach when reading *anything* that's written by a person you don't ultimately trust, i.e., yourself.

The latest Website is usually available at https://libre.andrewyu.org. The latest repository is usually available at https://git.andrewyu.org/libre.andrewyu.org.

Preface

Modern countries have laws that restrict the liberty of people in unacceptable ways but also don't guarantee the security of people at the same time. Governments are corrupted, inefficent, or both. This makes the lives of people miserable.

This experiment aims to develop the ideological base of a country with this main goal:

> The power of the government is limited to an extent where people have the freedom *from* persecution, intentional killing, and extreme poverty, while liberty *to* do what they will is not significantly compromised.

Choose a better word, persecution does not fit

This is an seemingly impossible task. However, considering the recent development of mathematics (especially in ideas such as group theory) and computer science (especially upcoming quantum computers), I believe there may be a algorithmic way to run a country.

Note the vague use of "run a country". This may mean a function of government decision over public opinion, the economy, expert opinion (along with a description of the expert). It might as well be a function of accepting the decision of the executive branch over the circumstances of the decision. The point is, there may be an referentially transparent function that decides how the country runs based on all the information we know.

A lot of ideas presented in this book are too vague. The mission of this project includes turning these abstractions into a solid system of society and law, where these are defined and leave minimum space for misconceptions and ambiguity.

In this book, we'll also identify problems in society, then present our solutions to it, if any. To the best of our ability, we'll provide examples and real-life stories in order to better explain the ideas. Then, we'll propose a legal framework for a country with the aforementioned goals. We'll try to make them specific and cover every possible case. Even though our math systems aren't complete (thanks to Gödel), we believe that in the field of law, things could (and should) be made mostly complete. Otherwise, it would be unclear what the country should do in an unexpected situation. Note that, if you find a problem with completeness over cases, or some scenarios where our theory creates results that are (in your opinion) bad or unacceptable, that is considered a bug. Please report them to the mailing list.

This book is part of the Libre Society project, available at https://project.andrewyu.org/libresociety.

This book, or the Libre Society project in general, is not endorsed or affiliated with the Free Software Foundation. However, we share many views, especially on copyright issues, and the rights people have on their computing.

The authors of this book aren't experts in mathematics, computer science, or politics. The intelligence of us authors combined are unlikely to yield a substantial result, for example, the aforementioned algorithm. Anyone who is interested in giving ideas, participating in revising the book, and/or contributing by other means is welcome! Please refer to page ix.

Nothing in this book is considered legal advice. We take no responsibility for your actions due to reading this book.

Community

Please write to our mailing list at libresociety@andrewyu.org. In order to join the list, send an email to the list.

We also have an IRC channel at `#libresociety` on `irc.andrewyu.org` and at `##libresociety` on `libera.chat`. These two channels are relayed to each other, you won't be able to see users on the other network, though. Users unfamiliar with IRC may use email or https://web.libera.chat/##libresociety.

Please note that, at this stage, the main means of discussion is the LibrePlanet Discuss mailing list, located at https://lists.libreplanet.org/mailman/listinfo/libreplanet-discuss. Many members there give precious suggestions to social, free software, and other related issues.

Resources

There are a few resources that are in this book's repository. Unless otherwise noted, they are not written by the authors of this book.

Abstract Algebra contains the basics of Abstract Algebra. It was fetched from https://math.berkeley.edu/ apaulin/AbstractAlgebra.pdf

Concrete Mathematics contains the basics of math in modern computer science. It was fetched from https://www.csie.ntu.edu.tw/ r97002/temp/Concrete%20Mathematics%202e.pdf

Das Kapital is a criticism of modern Capitalism by Karl Marx.

Contribution

By contributing to this work, you agree that your contributions are made available in the public domain and under the simple license on page ii.

If you're planning to give contributions related to your ideas directly to the code of the book, you could choose one of the following methods:

- Write to the mailing list describing your changes;
- Diff the file from the original. Git diffs are preferred but are not necessary;
- Clone from and push to `git://git.andrewyu.org/libresociety`, which isn't set up yet (your changes will be reviewed and merged to the main branch); Yes, remind Andrew to set it up.
- Become a coauthor, if you think you're contributing a lot.

If you have contributed very significant ideas to this project, we may invite you as a coauthor. You can also request to be a coauthor too. This section describes the technical way of how a coauthor contributes.

First of all, all coauthors would be added to the Git respository of this project, hosted at https://git.andrewyu.org.

You need to know how to add, commit and push with respect to Git bare repositories with SSH via SSH public key authentication. There are many guides online to this. Here we included a short guide.

```
# Obviously the numbers might change, and we assume you have Git installed and
# configured your Git name and email.  If you haven't, install \verb|git| from
# your package manager and read the manual page \verb|gittutorial|.)
~ $ git clone username@git.andrewyu.org:/var/www/git.andrewyu.org/libresociety
# Of course, replace the ‘username’ with the username we set for you.
Cloning into 'libresociety'...
remote: Enumerating objects: 17, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 17 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (17/17), 156.22 KiB | 120.00 KiB/s, done.
Resolving deltas: 100% (1/1), done.
~ $ cd ./libresociety/
~/libresociety $ git pull  # Make sure you have the laterst version.
# If an editor asks you for a commit message, just save and quit it.  If it
# talks about conflicts, you probably have to learn basic Git, i.e., how to
# resolve conflicts in merge.  I currently recommend against using rebase.
```

```
~/libresociety $ $EDITOR manifesto.tex
# If this does not work out for you, replace ‘$EDITOR’ with the name of your
# plain text editor of choice.  Do not use something like LibreOffice to edit
# this, of course.  Please read the comments (they start with ‘%’); they
# provide crucial information.
~/libresociety $ make
# This will fail if you do not have a proper LaTeX installation.  We strongly
# recommend installing one.  See [1] for a guide.  Use the ports tree if you
# are on OpenBSD.  The other BSDs and GNU/Linux distributions should have it
# available as a package.  On Alpine Linux, it is texlive-full.  On Arch
# Linux and its derivatives, it’s texlive-most in the repo and texlive-full
# in the AUR.  On Debian GNU/Linux and its derivatives, it is texlive-full.
# The packages above take a lot of disk space.  You could also just install
# core or minimal versions instead.  We try to keep our code simple, i.e.,
# by implementing the simpler macros that we need rather than calling a huge
# package.  If the minimal distribution doesn’t compile this, please give us
# the full log (that is, ‘manifesto.log’) and system information in the
# lists.  However, we do use bibliography management, so you would need a
# few extra packages.
[1] https://tug.org/texlive/doc/texlive-en/texlive-en.html#installation
... LaTeX output ...
~/libresociety $ git add manifesto.tex manifesto.pdf
# This step stages the changes you added.  Think of this as putting your files
# into a drawer.
~/libresociety $ git commit -m ’commit message’
# Replace ‘commit message’ with a short description describing what you
# changed.  Avoid commit messages like ‘edit’ or ‘update’ because they don’t
# give us any useful information.
# This step commits the staged files.  Think of this as scanning the files
# from your drawer in the archive system.  It will also digitally sign your
# commit if you configured Git to do so, which is recommemded.
~/libresociety $ git push
# This step updates the changes to the server, who in turn runs a Git hook to
# update the website (you don’t have to mind how this works).  Note that other
# coauthors need to run ‘git pull’ to ‘syncronize’ with this new version from
# the server, which will merge the changes.
```

Please also take a look at the style conventions in Appendix A.

Contents

Authors

Andrew Yu Website https://www.andrewyu.org, email andrew@andrewyu.org, irc Andrew on irc.andrewyu.org with SSL (WIP, doesn't work at the moment).

Part I

Problems in our Current Society

Chapter 1

Nonfree Software

(Andrew Yu)

1.1 What is Free Software?

As background, free software is software where the user is free to:

1. Run the program for any purpose;
2. Learn how the program works and/or modify the program to their needs;
3. Redistribute the original and/or modified program.

1.2 `systemd`, and Profit from Free Software

Critics of free software, who tend to call free software "open source" in order to remove the link to freedom and liberty, argue that it is impossible to profit developing free software, especially with copyleft licenses such as the GNU General Public License. The Free Software Foundation argues that software developers could still profit, since they are allowed to sell the software while redistributing. Common practice also includes paid support for systems that ought to be reliable. Even along with donations, people earn much less selling and supporting free software than leasing nonfree software. If we were to go for profit, developing nonfree software is obviously the way to go, but it is unjust.

There are indeed good ways to make money off free software. However, I argue that in perfection, profit must *not* be the incentive to anything. This is a working way to get paid making free software. However I'd argue that this payment is dangerous for many types of development.

Inconsistant use of pronouns

Chapter 2

Copyright

Part II

A Formal System of Society

Appendix A

Style Conventions

In order to keep this book consistant and understandable, these language and source code conventions should be followed.

1. General use of language
   (a)
2. Sectioning
   (a) Here, sectioning, unless otherwise specified, includes all structure layers, such as parts, chapters, sections, subsections, subsubsections, paragraphs, and subparagraphs. The paragraphs and subparagraphs mentioned in this list item refers to structuring inline titles with `\paragraph` and `\subparagraph`, not `\par` or text separated by empty lines. However, almost anywhere else, paragraphs mean logical paragraphs in terms of Linguistics, i.e.,
   (b) Section titles should be in Title Case, where all words are capitalized except for pronouns, articles, prepesitions, and similar language compoments. The first letter of a title must be uppercase, unless if it is a proper noun where uppercase is inappropriate.
   (c) Cross reference labels go immediatly after the TEX group containing the section name. For example, use `\section{Section Title}\label{section:section-title}`.
   (d) For numbered sections, i.e., these in the mainmatter and appendix, use .
3. Abbreviations, spacing, and punctuation
   (a) A single comment symbol must be used at end of lines where a space is inappropriate.
   (b) A non-breaking space must be used inside of people's names. For example, write `Andrew~Yu` instead of `Andrew Yu`, and for other places where a line break between these words is inappropriate, such as book names.

(c) A comma proceeds i.e., e.g., and similar abbreviations. These abbreviations have a dot in each of those letters and after the last one.

(d) A peroid proceeds Mr., Mrs., ans similar abbreviations.

(e) Most abbreviations in common English should have periods after them.

(f) Abbreviations after proper nouns should probably not have periods after them.

(g) For all abbreviations that end with a period and a space but are not at the end of a sentence, the space proceeding the period must be written as `\␣` (a space followed by a backslash) in order to stress that this is an inter-word space, not an inter-sentence space.

(h) For all sentences that end with a capitalized word, no matter if the word is fully UPPER CASE or is just Capitalized, the space after the period (unless if it is at the end of a paragraph, where a space must not be there) must be written as `\@` in order to stress that this is an inter-sentence space, not an inter-word space.

(i) For lists written as "a, b, [...], and/or/etc. z", a comma proceeds the last item before the connective.

(j) In the source code, for the style of the source code itself, interword spacing should be one space, while intersentence spacing should be two spaces. This convention does not take priority when in conflict with other conventions.

4. Fonts

(a) Use `\nameofbook` for book names, `\nameofperson` for name of persons, and `\emph` for emphatics.

(b) Use the `quote` environment for one-paragraph quotes. Use the `quotation` environment for multi-paragraph quotes.

(c) Use the `point` environment to illustrate a major point being discussed.

5. Source

(a) Do not load extra LaTeX packages unless necessary. If all is needed from a package is a simple macro, implement it in the preamble.

(b) Nested lists' environment declarations should be indented right under the text content of the parent item. Items inside the list environment and most environments in general should be indented two spaces from the start of the 'begin' control sequence, including the backslash.