NI 0.2.3 Manual

Michael Jumper

This manual is for NI version 0.2.3.

Table of Contents

1 Introduction

1.1 What is NI?

NI (an acronym for not-intractable, pronounced as the English "knee") is a handy boolean algebra program which has been released as free software under the **GNU General Public License**. It is capable of computing and manipulating truth tables for boolean functions involving hundreds (or thousands) of variables, even though your computer doesn't even remotely have enough memory or speed. How does it do this? Simple: *it cheats*.

1.2 What do you mean there are only 2^{250} particles in the universe?

Although truth tables are usually a convenient way to solve boolean logic problems, they rapidly become *inconvenient* as the number of variables increases (a truth table for n variables is 2^n in size). This is particularly bad for problems involving 250 or more variables, since not only is your computer not fast enough to finish such a calculation before the world ends, but the universe doesn't even contain enough particles to store the result. That's okay, though, because NI cheats: it *compresses* the result. In fact, the entire truth table is produced and manipulated in a compressed form. This is good for a number of reasons:

- The compressed form will fit in your computers memory.
- Manipulating a small piece of data is quicker than manipulating a large piece of data; your calculations will finish in a sane amount of time, rather than in billions of years.

1.3 But compression is a myth!

It is. You *cannot* losslessly make data smaller in some cases without making it larger in others — it simply isn't possible. NI will solve many problems in a much smaller amount of time, but, by the laws of information theory, there must be cases where NI's compressed form is equal to or larger than the original uncompressed truth table. In such a case, NI would probably be slower than a simple brute-force truth table method, but I have yet to find a function involving many variables that demonstrates this condition.

1.4 So what are NI's limitations?

NI makes use of a giant memory-speed tradeoff, so it is definitely possible to find a function which uses up all available memory on your computer — this is an unavoidable physical limitation of any computer program. Just because NI is capable of performing huge calculations quickly, doesn't mean that your computer has enough memory to perform these calculations; often, for complicated expressions involving hundreds of variables, the amount of memory required is several gigabytes. Keep in mind, however, that for hundreds of variables, a couple gigabytes of memory is nothing compared to the size of the uncompressed truth table.

2 NI Terminology

2.1 General Variables

General variables represent a single binary bit (`0` or `1`) of information and vary independently from each other. A general variable in NI essentially represents an unknown value, or all possible values of a single bit. All general variables are predefined in NI, and the syntax of their names is simple: the first general variable is `g0`, the second is `g1`, the third is `g2`, etc. For convenience, the lowercase letters `a` through `z` can be used in place of the first 26 general variables (`g0` through `g25`). Using N general variables together in an expression is logically equivalent (but not algorithmically equivalent) to calculating the entire truth table for that expression.

2.2 Truth Tables

NI uses a specialized form of truth table internally, but it can easily convert this form into a more human-readable, standard truth table using the `tt=` command.

```
NI> a & b & c
NI> tt=

 a b c | result
-------+-------
 0 0 0 | 0
 1 0 0 | 0
 0 1 0 | 0
 1 1 0 | 0
 0 0 1 | 0
 1 0 1 | 0
 0 1 1 | 0
 1 1 1 | 1

NI>
```

Note that the variable `a` (`g0`) changes its value every 2^0 bits, `b` (`g1`) changes its value every 2^1 bits, and `c` (`g2`) changes its value every 2^2 bits. This is true of all general variables: the general variable `gN` changes its value every 2^N bits.

2.3 Linear Truth Tables

Because the values of general variables follow a simple, reproducable pattern, the left side of a truth table (the side which lists the values of each variable involved in the function) can be left out while still preserving the content of the truth table. This form of a truth table is close to what NI uses internally, and is displayed on only one line, hence the name. The `ltt=` command can be used to display the linear truth table representing the previous result.

```
NI> a & b & c
NI> ltt=
00000001
NI>
```

2.4 Abbreviated Linear Truth Tables

Abbreviated linear truth tables (ALTTs) are the internal representation of truth tables used in NI. They are substantially harder to read than a standard truth table, but they take up much less memory. In an ALTT, repeating sections of a linear truth table are split into pieces with lengths that are a power of two (called "segments"). The notation of a segment is simple, either `0:n`, `1:n`, or `(ALTT):n`, where `ALTT` is any abbreviated linear truth table of length `n` or less, and `n` is an integer representing the exponent of a power of two. The `|` character is placed between two segments to denote that the segments are joined together. The `=` command is used to print the ALTT of the previous result.

```
NI> a & b & c
NI> =
0:2|(0:1|(0:0|1:0):1):2
NI>
```

The usefulness of ALTTs is more easily demonstrated with larger truth tables.

```
NI> g100 & g200
NI> =
0:200|(0:100|1:100):200
NI>
```

The truth table of the above calculation is beyond the capabilities of any computer to print, since it involves 2^{200} bits. If you attempt to print such a truth table, NI refuses with an error.

```
NI> g100 & g200
NI> =
0:200|(0:100|1:100):200
NI> tt=
ERROR: Physical result of operation is too large.
tt=
^
NI>
```

3 Examples

3.1 Validating and Invalidating Argument Forms

NI's built-in `assume` and `verify` commands can be used not only to apply hypothetical conditions, but also to validate arguments. In this section, we will verify three valid argument forms (modus ponens, modus tollens, and disjunctive syllogism) and verify the invalidity of two logical fallacies (affirming the consequent, and denying the antecedent).

3.1.1 Modus Ponens

The argument form of Modus Ponens is as follows:

> If P, then Q.
> P.
> Therefore, Q.

These can be translated into NI expressions quite easily:

```
p implies q
p
q
```

Now, to verify the validity, we need to use the `assume` command on each premise, and the `verify` command on the conclusion.

```
        NI (Not Intractable) version 0.2.3 (almost-stable)
        http://www.nongnu.org/not-intractable
        Distributed under the GNU General Public License.
        See the file COPYING for more information.
        Copyright (C) 2006 Michael Jumper

        Type quit to exit.

NI> assume (p implies q)
NI> assume (p)
NI> verify (q)
True.
```

NI agrees that this is a valid argument.

3.1.2 Modus Tollens

The argument form of Modus Tollens is as follows:

> If P, then Q.
> Q is false.
> Therefore, P is false.

The corresponding equivalents in NI are:

```
p implies q
q == 0
```

```
p == 0
```

And now for the verification:

```
NI> reset
NI> assume (p implies q)
NI> assume (q == 0)
NI> verify (p == 0)
True.
```

NI agrees that this argument form is valid. Because there were conditions left over from the previous argument form verification (the modus ponens example), we needed to use the `reset` command to remove those conditions.

3.1.3 Disjunctive Syllogism (Modus Tollendo Ponens)

The argument form of Disjunctive Syllogism is as follows:

P or Q.
Not P.
Therefore, Q.

Translated into NI expressions:

```
p | q
!p
q
```

And now for the verification:

```
NI> reset
NI> assume (p | q)
NI> assume (!p)
NI> verify (q)
True.
```

Once again, NI agrees with the validity of this argument form.

3.1.4 Affirming the Consequent

The form of the logical fallacy Affirming the Consequent is as follows:

If P, then Q.
Q.
Therefore, P.

The NI expressions:

```
p implies q
q
p
```

And now for the actual test:

```
NI> reset
NI> assume (p implies q)
NI> assume (q)
NI> verify (p)
False.
```

As you can see, NI is also capable of showing argument forms to be invalid. This particular logical fallacy is fallacious because it assumes that the "if ... then ..." goes both ways, when it does not. The argument would be valid if if-and-only-if were used instead (as this is a bidirectional if):

```
NI> reset
NI> assume (p iff q)
NI> assume (q)
NI> verify (p)
True.
```

3.1.5 Denying the Antecedent

The form of the logical fallacy Denying the Antecedent is as follows:

If P, then Q.
P is false.
Therefore, Q is false.

The NI expressions:

```
p implies q
p == 0
q == 0
```

And now for the actual test:

```
NI> reset
NI> assume (p implies q)
NI> assume (p == 0)
NI> verify (q == 0)
False.
```

Indeed, this argument form is false.

4 Optimization

4.1 The Concept

NI is faster than other truth-table based algorithms because it solves a common problem in a different way. The same can be said about certain expressions given to NI; one particular expression can likely be rewritten to solve the same problem, yet run faster. NI takes care of a large amount of the optimization necessary to solve problems with large numbers of variables, but still some effort must be made to lessen NI's workload wherever possible.

4.2 An Example

Lets say we want to XOR together 256 variables, and calculate the corresponding truth table. In NI, this is as simple as typing the expression (or having another program generate it for you). Because of the properties of XOR, it doesn't matter *logically* what order the variables are placed within the expression, but just for the sake of experiment, lets benchmark NI on the same expression in two different orders.

4.2.1 256 XORs: Order 1

First, we'll try XORing the variables together with the numbers of each general variable decreasing from `g255` to `g0` (the entire expression will not fit nicely on one page, so only a piece is shown below):

```
g255 ^ g255 ^ g254 ^ g253 ^ g252 ^ g251 ^ g250 ^ g249 ^ ...
```

Timing how long it takes NI to calculate the result shows that, for this particular problem, NI needs **0.250 seconds**.

4.2.2 256 XORs: Order 2

Now, lets try reversing the order, such that the general variable numbers are increasing from `g0` to `g255`:

```
g0 ^ g1 ^ g2 ^ g3 ^ g4 ^ g5 ^ g6 ^ g7 ^ ...
```

Timimg this reveals that NI requires only **0.006 seconds** to solve the *exact same problem* when the order is reversed.

4.2.3 Whats going on?

Because of how NI solves problems, having the general variables decrease in number from left to right forces NI to move further recursively into the ALTT than it would have to if the variables were increasing in number from left to right. For most expressions, this is true; therefore, in general, if you do not care what general variable is used to represent which variable of your function, try to keep the general variable numbers increasing from left to right. For example, use `a | (b & c)` rather than `c | (b & a)` or `(a | b) & (a | c)`. To be more specific: try to keep complicated results small in length, and simple results or values (like general variables) larger in length. Doing so will reduce the amount of work that NI must perform to accomplish your task.

Appendix A Syntax Reference

A.1 Comments

Comments begin with a `#` symbol and are effective until the end of the current line.

A.2 Operators

Operators take two arguments: the first argument is simply the previous result, the second is the value which follows the operator. The overall syntax is `OPERATOR VALUE`.

`!&` NAND

`|` OR (implemented through NAND)

`&` AND (implemented through NAND)

`^` XOR (faster than the NAND version)

`^^` XOR (implemented through NAND)

`=>` Store result in the following named variable.

`undef` Delete (undefine) the named variable.

`==` Compares two values. Result has 1's where equal, 0's where not equal. Equivalent to the inverse of an XOR.

`!=` Compares two values. Result has 0's where equal, 1's where not equal. Equivalent to XOR.

`implies` Logical implication / if ... then

`iff` if-and-only-if (implemented with non-NAND XOR)

`iff2` iff (implemented through NAND)

A.3 Values

Values are the arguments of operators, but can exist on their own as the entire expression.

`(EXPRESSION)`
A parenthesized expression.

`!VALUE` The inverse of VALUE. Something like `!!a` is legal but useless.

`gN` General variable. Replace N with an integer `>=` 0. General variables are taken to be unknown and independant from each other.

`a` Any lowercase letter is an alias for g0 through g25 (where `a` = `g0`, `b` = `g1`, `c` = `g2`, etc.)

`$VARNAME` Named variable.

`0` Binary 0 / Contradiction

`1` Binary 1 / Tautology

`?` An unknown value which may be 0 or 1.

`$ANS` The last result (ANSwer from the previous line).

`$MASK` The mask created by direct assignment or through the assume command.

A.4 Commands

Commands are like operators but they do not operate on the last result.

`info` Print statistical information.

`verify` Takes one parameter. If the parameter is universally true or true because of an assume statement, prints "True". Otherwise prints "False". Returns nothing.

`assume` Takes one parameter. The parameter is used as a hypothetical statement, and all calculations and verifications thereafter are restricted by the parameter. Returns nothing.

`reset` Removes all previous assumptions. Equivalent to `undef $MASK`.

`simplify` Returns a simplified version of the previous result based on `$MASK`.

A.5 Printing Results

NI provides several ways of printing the last result; each prints the result in a different format.

`=` Print the last result as an ALTT[1].

`tt=` Print the last result as a truth table

`ltt=` Print the last result as an LTT

`id=` Print a unique number representing the result (a memory address, actually). Equivalent results will have equivalent ids.

`f=` Print a function which has a result that is logically equivalent to the previous result.

[1] Results which involve many variables may only be able to be printed as ALTTs because of sheer size

Appendix B Invoking NI

B.1 Syntax

`ni [OPTIONS] [FILE [PARAMETERS[1]]]`

B.2 Options

`-v` Increases the verbosity level by one. Repeating this option increases the verbosty even further.

`-sN` Set the simplification to N, where N is an integer from 0 to 2. This only affects the simplify and f= commands. The current default is -s1. -s2 is experimental and fails for some expressions.

`-h, --help`
Display information on usage.

`--print-large`
By default, NI will refuse to print truth tables for functions involving more than 26 variables, even though the calculations have been performed, returning an error referring to the size of the result. This option overrides that error, and forces NI to print such truth tables anyway. Use with caution. description

[1] If a file is specified, all remaining arguments on the command line will be passed to the file itself as the variables $1, $2, $3, ... etc.

Appendix C Errors

If you happen to get an error, it is usually related to syntax or spelling. Double check the syntax reference (included in the appendices). If you believe you have found a bug, see the section entitled Reporting Bugs.

`ERROR: Variable name expected.`
: You typed an operator which takes a named variable as an argument (like `$my_variable`), but did not provide the named variable.

`ERROR: Value expected.`
: You typed an operator, but did not provide the required argument. A misspelled command or operator will also cause this error.

`ERROR: Variable undefined.`
: You typed a named variable, but that variable hasn't been defined yet.

`ERROR: Physical result of operation is too large.`
: You attempted to print the last result as a truth table or linear truth table, but the last result is too large to be printed this way. Try the `info` command to see how large the result is, or restart NI with the `--print-large` command line option to override this error (if necessary).

`ERROR: Could not open input file: FILENAME`
: The file provided on the command line does not exist; check the spelling, check the current working directory, etc.

`ERROR: [undefined error number... this shouldn't be happening]`
: If you *ever* get this error, report it immediately! You've found a bug!

C.1 Reporting Bugs

If you believe you have found a bug, send a descriptive e-mail (and a copy of the script that reproduces the bug, if possible) to not-intractable@nongnu.org.

Appendix D Copying This Manual

D.1 GNU Free Documentation License

Version 1.2, November 2002

0. PREAMBLE

   The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

   This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

   We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

   This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

   A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

   A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

   If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

   If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

   It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

   You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

   A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
H. Include an unaltered copy of this License.
I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

   You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

   The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

   In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

D.1.1 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Index

Command and Operator Index