Evolutionary Iterated Prisoner's Dilemma

An approach for finding good strategies

Version 0.2

Marko Grönroos (magi@iki.fi)

August 13th 2003

About this document

This document provides information about the evolutionary approach to Iterated Prisoner's Dilemma.

Copyright and License

Table of Contents

Chapter 1 Introduction

Prisoner's dilemma is a classical game-theoretic model of cooperative decision-making. It is very interesting from the viewpoint of sociobiology, psychology, economics, politics, and many other fields of science. Many books and hundreds of articles have been written about it since 1970s. In sociobiology, it has been used in the research of evolution of cooperative behavior.

The game is very simple. Two suspects have been arrested for possession of stolen goods and they are suspected of burglary. Each is proposed with a deal: confess the burglary (*defect* your partner) and walk away free, while your partner will get five years. However, if you don't confess (you *cooperate* with your partner), and your partner defects, you will get five years and your partner gets freed. If neither confesses (both cooperate), both get two years in prison (for possession of stolen goods) and if both confess (defect), both get three years.

Below is a "reward" matrix for the game:

		Partner's decision	
		COOPERATE	DEFECT
Your decision	COOPERATE	Both get 1 year	You get: 5 years Partner gets: nothing
	DEFECT	You get: nothing Partner gets: 5 years	Both get 3 years

If the game is played only once, it is statistically better to defect, as the partner's decision can not be predicted in any way. Things get more interesting in an iterated version of prisoner's dilemma (IPD), where the game is played for several rounds and the players can make their choice according to the previous choices made by the partner. For example, if your partner defected in the first round, you might think he will defect also in the next round. In our version of the IPD, the players remember the choices made in three previous rounds.

Players (prisoners) can choose various strategies. For example, they can choose to always defect or cooperate. Tit-for-tat is a well-known strategy where player cooperates on the first round, and then mimics the choice made by the other player. Tit-for-tat is the optimal strategy for playing against a partner playing tit-for-tat, and it's good against an always-defecting strategy, but it's not optimal against an always-cooperating strategy, nor a completely random strategy. In a tournament of different strategies, the optimal strategy always depends on the set of other strategies.

Using evolutionary algorithms for finding strategies for IPD has been done by many people and this approach is just one. Some studies have competed the population of evolved solutions against each other and used variable length of memory of previous moves. In this implementation, we evolve a solution that competes against a fixed set of trivial strategies. The implementation uses a fixed memory length of three previous rounds, because it is easy to implement with a fixed length genome, and doesn't require any complex recombination techniques.

EvoIPD uses the NeHeP evolutionary algorithm library, which uses the MagiC++ base class library for C++. Both EvoIPD and NeHeP are provided in MagiC++ source package. EvoIPD is provided as a "project" application for NeHeP in the `libnhp/projects/prisoners` subdirectory of the MagiC++ source tree.

1.1. System requirements

MagiCBuild has the following system requirements:

- GNU/Linux operating system
- GNU Make 3.79.1 or newer
- Doxygen (optional)

Platforms

The following Linux distributions have been tested:

Distribution	
Red Hat Linux 9	
Mandrake 9.1	MagiC++ does not compile, problem with va_list.
Debian 2.2 + upgrades	MagiC++ does not compile, problem with va_list.

1.2. Licensing

Prisoners is licensed under the *GNU General Public License* (GPL). The GNU General Public License is given in file `docs/COPYING`.

This documentation is licensed under the *GNU Free Documentation License*, as presented in the Chapter 5.

Chapter 2 Installing

This chapter provides instructions for configuring, building, and actually installing EvoIPD and its required libraries.

EvoIPD is provided as a part of the MagiC++ library package, which includes also the NeHeP library. The EvoIPD is a "project" application for NeHeP.

The installation instructions for the software package would probably be very much like the instructions in this chapter.

Note: *You are granted a permission to use any material in this (and only this) chapter for the installation instructions of your software. We hereby waive the copyrights for the contents of this chapter.*

2.1. Opening the source package

The source code is normally provided as a GNU Tar package compressed with `bz2`. You can unpack it with the following shell command:

```
tar jxf magic++-1.3beta1.tar.bz2
```

This will unpack the source code into an appropriate subdirectory under the current directory.

2.2. Configuring

To configure the source code for compilation, change to the source directory and run the `configure` script as follows:

```
cd magic++-1.3beta1
./configure
```

Optionally, if you wish to later install the package (headers and library) to some other than the default directory, you need to set the installation path with the `--prefix` attribute:

```
./configure --prefix=/opt/magic++
```

The default path for *root* user is `/usr/local`, and for other users their home directory.

No other configuration flags are currently supported.

2.3. Compiling

Include dependencies have to be determined before actual compiling, with the following command:

```
make deps
```

This may produce some errors, which are usually not relevant. Making dependencies is important if you intend to recompile the sources after making changes to them.

The package is compiled with the following simple command:

```
make
```

2.3.1. Compilation output

The output binaries as well as any intermediate files of the compilation will be located in an output directory tree separate from the source tree.

The build framework does the compilation output in separate directory, determined by the configuration script. The default output directory is located in:

`/tmp/$USER/build/`<*architecture*>`/release`

where `$USER` is the user name and *architecture* is the operating system and processor architecture, for example, `Linux-i686`.

For example, binaries are found under the `bin` subdirectory:

```
cd /tmp/$USER/build/Linux-i686/release/bin
./evoipd
```

Yoy can clean the output with the following command in the top-level source directory:

```
make clean
```

You do not normally need to clean the output.

2.4. Installing

After compiling, you can be install the package under the configured installation directory (see above) by issuing the following command in the source directory:

```
make install
```

This will copy the output library binaries and header files to appropriate subdirectories under the installation directory.

Directory	***Description***
`<`*`instdir`*`>/lib`	Libraries
`<`*`instdir`*`>/bin`	Binaries
`<`*`instdir`*`>/include`	Header files

2.5. Uninstalling

You can remove the installation by giving the following command in the source directory:

```
make uninstall
```

This removes the installed files and directories only if the installation path has not been changed with `configure` script after installing.

Chapter 3 Software architecture

3.1. NeHeP evolutionary library

NeHep is a highly object oriented general-purpose evolutionary algorithm library for C++. It is based on the MagiC++ base class library for C++.

In conventional genetic algorithms, the genetic code is typically just a bit string or array of real values. The bit string is decoded by some procedure and then evaluated using a fitness function. NeHeP uses approach where the genetic code is potentially a very complex data structure. For all elements of that structure, there must be an implementation of the basic genetic operators, recombination and mutation. When such operators are applied to the top-level container (a genome), they are recursively applied to all lower-level genetic structures, and any parameters (such as mutation rates) can be modified by the specific implementations of the operators at each structural level. Another, a rather agent-oriented, feature is that, although the genetic code can be read as usual by a "decoding function", the genetic structures (genes and genetic containers) can be sent a message that "activates" them and they build the phenotype of their host individual just by themselves, possibly by activating other genes. Genotypic features can be designed in a modular manner, and then just "thrown in" in the genome, and the individual's ontogenetic procedure will take care of the "growth process".

3.2. EvoIPD architectural overview

EvoIPD implements two classes required by NeHeP's programming framework: an evolution environment and a genetic container that contains the genes to be evolved.

The EvoIPD evolution environment, ***PrisonEAEnv***, is an application-specific implementation of the abstract ***EAEnvironment*** class, which describes the actual problem for which the evolutionary algorithm tries to find solutions. It has two tasks. First, it has to initialize the genome of the individuals in the population with problem-specific genetic structure. Second, it provides an evaluation function for evaluating the fitness of a given genome, or individual, in the environment (problem).

The ***PrisonerGene*** is a genetic container (***Gentainer***) that contains the genetic material that represents an "encoded" solution. The ***PrisonEAEnv*** initializes the decoding of the genome by sending it the identifier of the ***PrisonerGene***, "PS". When the gene receives this message, it decodes itself as a phenotypic feature of the individual. The phenotype is actually a rule string for the IPD game, which the gene sets as an attribute for the individual. ***PrisonEAEnv*** reads this attribute, uses it to create a ***PDStrategy*** instance, and plays the IPD game with it against all the other strategies. The evaluation function returns (reverse of) the game score as the fitness of the individual. The evolutionary algorithm uses this value to guide the evolution.

Figure 1 below illustrates the class relationships in EvoIPD.

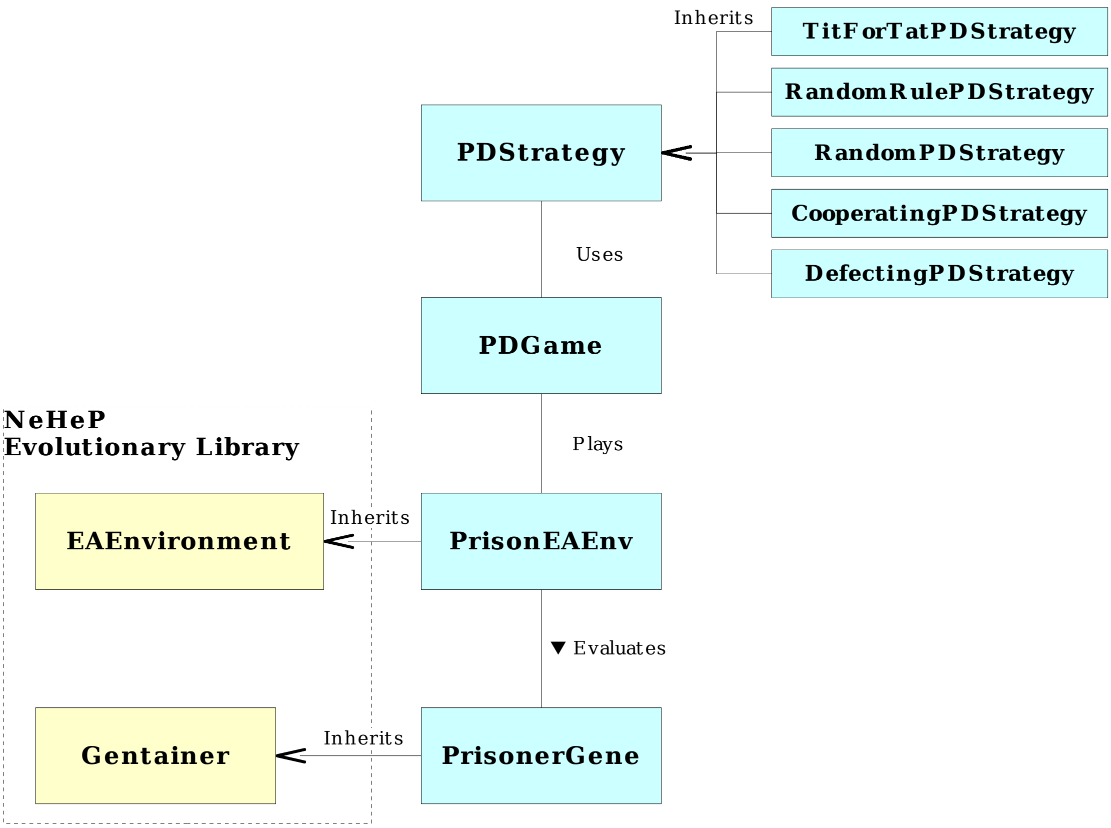

Figure 1: Class relationship diagram for EvoIPD

3.3. Competing strategies

The hand-made strategies are as follows:

Cooperating A "dumb" strategy that always cooperates disregarding whatever wrongs the partner has done previously. It performs quite nicely against cooperating and tit-for-tat partners, but is completely at mercy of the Defecting strategy.

Defecting A greedy and untrusting strategy that always defects disregarding whatever the partner does. This strategy is best for non-iterated prisoner's dilemma, but not for iterated. While it can, of course, abuse the unsuspecting Cooperating strategy, and does optimally against random strategies, it does very badly against tit-for-tat.

TitForTat A clever strategy that does very well against most strategies. On first round, it cooperates, and on successive rounds, it does what the partner did in the previous round. Tit-for-tat was first proposed in 1970s by Axelrod for a competition of human-designed game strategies. It won all the more complex strategies. However, it does not do very well against random strategies, and it can not abuse the dumb cooperating strategy.

Random A nut strategy that makes every decision randomly. As it is hard to predict its move, it has (exactly) average success against any strategies that try to predict it. Only determinedly defecting strategies do somewhat well against it.

RandomRule As Random above, except that the rules are set randomly in the beginning of the game. A learning strategy could, in principle, learn to abuse the RandomRule strategy. However, the strategy is currently re-initialized in the beginning of each game, so the strategy is completely random from the viewpoint of the evolutionary algorithm, which can therefore not learn to play against it.

Hundreds of other human-designed strategies exist for the iterated prisoner's dilemma. One very annoying strategy is one that first cooperates, and cooperates if the partner does, but avenges any defection by defecting indefinitely. While some clever strategies could "test" other strategies to see how gullible they are, and thus surpass Tit-for-tat, the existence of vengeful strategies in a tournament quickly turns the tables against such naughty strategies.

3.3.1. Representation of strategies

The ***PDStrategy*** class allows the representation of a strategy as a rule table. The rule table has six parameter columns: two for each step of "memory" in the past rounds. The two columns are for the decisions made by self and the partner. These six columns can have 64 possible binary combinations of "defect" and "cooperate".

In addition, each strategy has six "hypothetical" or "prehistorical" items, which represent imaginary choices made by both players before the game actually begun. Without these items, we would need to have separate rules for the beginning of game.

An example of a rule table evolved by the algorithm can be seen below in Section 4.2 *Results*.

Chapter 4 Experiments

4.1. Parameters

Population size of 20 individuals was used. Auto-adaptation of mutation rates was used with initial mutation rate of p=0.1 and lower bound of p=0.01 for binary genes. Recombination frequency of p=0.5 was used.

4.2. Results

The evolutionary algorithm was run for 200 generations and the fittest individual was picked for final testing. Table below presents the score matrix of the different strategies.

	EvoIPD	***Tit4Tat***	***RandomRule***	***XRandom***	***Cooping***	***Defecting***	***Avg***
EvoIPD	1.00 \ 1.00	1.00 \ 1.00	1.41 \ 3.5	1.53 \ 3.93	1.00 \ 1.00	3.02 \ 2.97	*1.492*
Tit4Tat	1.00 \ 1.00	1.00 \ 1.00	2.24 \ 2.22	2.26 \ 2.24	1.00 \ 1.00	3.02 \ 2.97	*1.753*
RandomRule	3.89 \ 1.46	2.19 \ 2.21	2.24 \ 2.24	2.26 \ 2.46	0.52 \ 2.92	4.02 \ 1.47	*2.520*
Xrandom	3.94 \ 1.52	2.23 \ 2.26	2.25 \ 2.26	2.26 \ 2.26	0.51 \ 2.97	3.99 \ 1.51	*2.530*
Cooping	1.00 \ 1.00	1.00 \ 1.00	3.07 \ 0.48	3.00 \ 0.50	1.00 \ 1.00	5.00 \ 0.00	*2.346*
Defecting	2.97 \ 3.02	2.97 \ 3.02	1.51 \ 4.00	1.52 \ 3.99	0.00 \ 5.00	3.00 \ 3.00	*1.993*

The evolved solution clearly performed best, also better than Tit4Tat. Looking at the results more closely we can analyze the benefits of the evolved solution more clearly. It performed equally well against Cooperating and Defecting strategies as Tit4Tat did, but it performed much better against the random strategies. The optimal strategy against a random strategy would be to always defect, the same as in a single-round prisoner's dilemma, as having experience about previous moves does not help at all. As we can see from the table, the average score of the defecting strategy against both versions of random strategy is 1.5, as we can predict with (0+3)/2=1.5. We can therefore say that the evolved solution has learned to recognize random strategies very well.

The genotype of the best individual in the above experiment is given below:

```
Genome {PS=PrisonerGene
{0011000101011100101011010100110110110111110011111111101000111
01000000 Px=0.50 RM=1 Nx=1 } Px=0.50 RM=1 Nx=1 }
```

The semantics of the genotype are more clear in tabular format:

	Round								***Round***						
	-3		***-2***		***-1***				***-3***		***-2***		***-1***		
	Other	***Me***	***Other***	***Me***	***Other***	***Me***	***Decision***		***Other***	***Me***	***Other***	***Me***	***Other***	***Me***	***Decison***
0	Coop	Coop	Coop	Coop	Coop	Coop	*0: Coop*	32	Defect	Coop	Coop	Coop	Coop	Coop	*1: Defect*
1	Coop	Coop	Coop	Coop	Coop	Defect	*0: Coop*	33	Defect	Coop	Coop	Coop	Coop	Defect	*0: Coop*
2	Coop	Coop	Coop	Coop	**Defect**	Coop	***1: Defect***	34	Defect	Coop	Coop	Coop	**Defect**	Coop	***1: Defect***
3	Coop	Coop	Coop	Coop	**Defect**	Defect	***1: Defect***	35	Defect	Coop	Coop	Coop	**Defect**	Defect	***1: Defect***
4	Coop	Coop	Coop	Defect	Coop	Coop	*0: Coop*	36	Defect	Coop	Coop	Defect	Coop	Coop	*0: Coop*
5	Coop	Coop	Coop	Defect	Coop	Defect	*0: Coop*	37	Defect	Coop	Coop	Defect	Coop	Defect	*1: Defect*
6	Coop	Coop	Coop	Defect	**Defect**	Coop	***0: Coop***	38	Defect	Coop	Coop	Defect	**Defect**	Coop	***1: Defect***
7	Coop	Coop	Coop	Defect	**Defect**	Defect	***1: Defect***	39	Defect	Coop	Coop	Defect	**Defect**	Defect	***1: Defect***
8	Coop	Coop	Defect	Coop	Coop	Coop	*0: Coop*	40	Defect	Coop	Defect	Coop	Coop	Coop	*1: Defect*
9	Coop	Coop	Defect	Coop	Coop	Defect	*1: Defect*	41	Defect	Coop	Defect	Coop	Coop	Defect	*1: Defect*
10	Coop	Coop	Defect	Coop	**Defect**	Coop	***0: Coop***	42	Defect	Coop	Defect	Coop	**Defect**	Coop	***0: Coop***
11	Coop	Coop	Defect	Coop	**Defect**	Defect	***1: Defect***	43	Defect	Coop	Defect	Coop	**Defect**	Defect	***0: Coop***
12	Coop	Coop	Defect	Defect	Coop	Coop	*1: Defect*	44	Defect	Coop	Defect	Defect	Coop	Coop	*1: Defect*
13	Coop	Coop	Defect	Defect	Coop	Defect	*1: Defect*	45	Defect	Coop	Defect	Defect	Coop	Defect	*1: Defect*
14	Coop	Coop	Defect	Defect	**Defect**	Coop	***0: Coop***	46	Defect	Coop	Defect	Defect	**Defect**	Coop	***1: Defect***
15	Coop	Coop	Defect	Defect	**Defect**	Defect	***0: Coop***	47	Defect	Coop	Defect	Defect	**Defect**	Defect	***1: Defect***
16	Coop	Defect	Coop	Coop	Coop	Coop	*1: Defect*	48	Defect	Defect	Coop	Coop	Coop	Coop	*1: Defect*
17	Coop	Defect	Coop	Coop	Coop	Defect	*0: Coop*	49	Defect	Defect	Coop	Coop	Coop	Defect	*1: Defect*
18	Coop	Defect	Coop	Coop	**Defect**	Coop	***1: Defect***	50	Defect	Defect	Coop	Coop	**Defect**	Coop	***1: Defect***
19	Coop	Defect	Coop	Coop	**Defect**	Defect	***0: Coop***	51	Defect	Defect	Coop	Coop	**Defect**	Defect	***1: Defect***
20	Coop	Defect	Coop	Defect	Coop	Coop	*1: Defect*	52	Defect	Defect	Coop	Defect	Coop	Coop	*1: Defect*
21	Coop	Defect	Coop	Defect	Coop	Defect	*1: Defect*	53	Defect	Defect	Coop	Defect	Coop	Defect	*1: Defect*
22	Coop	Defect	Coop	Defect	**Defect**	Coop	***0: Coop***	54	Defect	Defect	Coop	Defect	**Defect**	Coop	***0: Coop***
23	Coop	Defect	Coop	Defect	**Defect**	Defect	***1: Defect***	55	Defect	Defect	Coop	Defect	**Defect**	Defect	***1: Defect***
24	Coop	Defect	Defect	Coop	Coop	Coop	*0: Coop*	56	Defect	Defect	Defect	Coop	Coop	Coop	*0: Coop*
25	Coop	Defect	Defect	Coop	Coop	Defect	*1: Defect*	57	Defect	Defect	Defect	Coop	Coop	Defect	*0: Coop*
26	Coop	Defect	Defect	Coop	**Defect**	Coop	***0: Coop***	58	Defect	Defect	Defect	Coop	**Defect**	Coop	***0: Coop***
27	Coop	Defect	Defect	Coop	**Defect**	Defect	***0: Coop***	59	Defect	Defect	Defect	Coop	**Defect**	Defect	***1: Defect***
28	Coop	Defect	Defect	Defect	Coop	Coop	*1: Defect*	60	Defect	Defect	Defect	Defect	Coop	Coop	*1: Defect*
29	Coop	Defect	Defect	Defect	Coop	Defect	*1: Defect*	61	Defect	Defect	Defect	Defect	Coop	Defect	*1: Defect*
30	Coop	Defect	Defect	Defect	**Defect**	Coop	***0: Coop***	62	Defect	Defect	Defect	Defect	**Defect**	Coop	***0: Coop***
31	Coop	Defect	Defect	Defect	**Defect**	Defect	***1: Defect***	63	Defect	Defect	Defect	Defect	**Defect**	Defect	***1: Defect***

Extracting the logic of the solution from the decision table is not trivial and I will not do it here. We can, however, try to analyze the strategy with some simple statistics of the decisions.

38 (59%) of 64 decisions are defecting, which makes the strategy more likely to defect than to cooperate for a given random history. The cases where the partner defected on previous round are emphasized in bold face; 18 (56%) of 32 such decisions are defecting, so the strategy does not resemble Tit4Tat (which would always defect in such cases) very much. In fact, the strategy seems to favor its own previous decision more often, in 39 (61%) of 64 cases.

If we average the number of defective decision in history for each defective decision, for example "I defected because: (I defected (1) + I defected (1) + I cooped (0))/3 = 2/3", we get total sums of 64/3 (56%) own defections of 38 and 56/3 (49%) other's defections of 38 defective decisions. This is comparable with the above ratio for the previous round.

While it is interesting to notice that the strategy doesn't resemble Tit4Tat much, it is difficult to say what these statistics might indicate. Closer analysis of the decision logic would be required.

As the last six values in the genotype are zeros, the strategy assumes that any hypothetical (prehistorical) decisions were cooperative.

4.2.1. Other experiments

The results above were from a single run, but it may.

In some other runs, the evolved strategy learned to abuse the cooperating strategy and got score 0.01 against it. In doing so, it however suffered slightly against all the other strategies, and the average score was roughly par with the score for the experiment presented above. It obviously had to defect once to distinguish between the cooperating and Tit4Tat strategies, and cooperate once to recognize the defecting strategy. As these detections take much of the complexity allowed by the rule table, it probably could not learn to recognize the random strategies so well.

4.2.2. Effectiveness of evolution

All experiments for studying the usefulness of evolution for problem-solving should be validated against random search. The evolution is supposed to sample the search space in a correlated manner, using the previously found solutions as a starting point for finding even better solutions. An evolutionary algorithm with a population of 20 and evolution length of 200 generations samples 4,000 points from the search space. A random search of 4,000 samples should not do better than that. The random samples must be selected with the same distribution, that is, with the same algorithm as the initial population for the evolutionary algorithm.

I validated the effectiveness of the evolutionary algorithm by setting the population size to 4,000 and running the "evolution" for one generation. The best found random strategy was actually quite good, with average score of 1.582, just slightly worse than for the evolved strategy. The best random strategy was an abuser that abused the cooperating strategy, while playing rather nicely with Tit4Tat and appropriately defecting the defecting strategy. The average score of all random strategies was 2.222 and the score for the worst strategy was 2.747.

4.3. Summary of the results

The evolved solution performed better than any of the predefined strategies. Other evolved solutions showed remarkable ability to recognize the other strategies by deliberately defecting and cooperating to see their response.

The results were not very reliable, as the experiment was run only once. The validity of the effectiveness of the algorithm is not quite certain, as the random search produced a rather good solution. The search space was sampled randomly only once; we do not know if the best found random solution was exceptionally good or bad for our sample size.

Chapter 5 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.

Alphabetical Index