Intuitionistic type theory

en.wikipedia.org

March 20, 2022

Contents

1 Intuitionistic type theory

Alternative foundation of mathematics **Intuitionistic type theory** (also known as **constructive type theory**, or **Martin-Löf type theory**) is a type theory[1] and an alternative foundation of mathematics[2]. Intuitionistic type theory was created by Per Martin-Löf[3], a Swedish[4] mathematician[5] and philosopher[6], who first published it in 1972. There are multiple versions of the type theory: Martin-Löf proposed both intensional[7] and extensional[8] variants of the theory and early impredicative[9] versions, shown to be inconsistent by Girard's paradox[10], gave way to predicative[11] versions. However, all versions keep the core design of constructive logic using dependent types.

1.1 Design

Martin-Löf designed the type theory on the principles of mathematical constructivism[12]. Constructivism requires any existence proof to contain a "witness". So, any proof of "there exists a prime greater than 1000" must identify a specific number that is both prime and greater than 1000. Intuitionistic type theory accomplished this design goal by internalizing the BHK interpretation[13]. An interesting consequence is that proofs become mathematical objects that can be examined, compared, and manipulated.

Intuitionistic type theory's type constructors were built to follow a one-to-one correspondence with logical connectives. For example, the logical connective called implication ($A \Longrightarrow B$) corresponds to the type of a function ($A \to B$). This correspondence is called the Curry–Howard isomorphism[14]. Previous type theories had also followed this isomorphism, but Martin-Löf's was the first to extend it to predicate logic[15] by introducing dependent types[16].

---

1 https://en.wikipedia.org/wiki/Type_theory
2 https://en.wikipedia.org/wiki/Foundations_of_mathematics
3 https://en.wikipedia.org/wiki/Per_Martin-L%C3%B6f
4 https://en.wikipedia.org/wiki/Sweden
5 https://en.wikipedia.org/wiki/Mathematician
6 https://en.wikipedia.org/wiki/Philosopher
7 https://en.wikipedia.org/wiki/Intensional_logic
8 https://en.wikipedia.org/wiki/Extensionality
9 https://en.wikipedia.org/wiki/Impredicative
10 https://en.wikipedia.org/wiki/Girard%27s_paradox
11 https://en.wikipedia.org/wiki/Predicativity
12 https://en.wikipedia.org/wiki/Mathematical_constructivism
13 https://en.wikipedia.org/wiki/BHK_interpretation
14 https://en.wikipedia.org/wiki/Curry%E2%80%93Howard_isomorphism
15 https://en.wikipedia.org/wiki/Predicate_logic
16 https://en.wikipedia.org/wiki/Dependent_types

1.2 Type theory

Intuitionistic type theory has 3 finite types, which are then composed using 5 different type constructors. Unlike set theories, type theories are not built on top of a logic like Frege's[17]. So, each feature of the type theory does double duty as a feature of both math and logic.

If you are unfamiliar with type theory and know set theory, a quick summary is: Types contain terms just like sets contain elements. Terms belong to one and only one type. Terms like $2+2$ *and* $2\cdot 2$ *compute ("reduce") down to canonical terms like 4. For more, see the article on Type theory[18].*

1.2.1 0 type, 1 type and 2 type

There are 3 finite types: The **0** type contains 0 terms. The **1** type contains 1 canonical term. And the **2** type contains 2 canonical terms.

Because the **0** type contains 0 terms, it is also called the empty type[19]. It is used to represent anything that cannot exist. It is also written $\bot$ and represents anything unprovable. (That is, a proof of it cannot exist.) As a result, negation[20] is defined as a function to it: $\neg A := A \to \bot$.

Likewise, the **1** type contains 1 canonical term and represents existence. It also is called the unit type[21]. It often represents propositions that can be proven and is, therefore, sometimes written $\top$.

Finally, the **2** type contains 2 canonical terms. It represents a definite choice between two values. It is used for Boolean values[22] but *not* propositions. Propositions are considered the **1** type and may be proven to never have a proof (the **0** type), or may not be proven either way. (The Law of Excluded Middle[23] does not hold for propositions in intuitionistic type theory.)

1.2.2 Σ type constructor

Σ-types contain ordered pairs. As with typical ordered pair (or 2-tuple) types, a Σ-type can describe the Cartesian product[24], $A \times B$, of two other types, A and B. Logically, such an ordered pair would hold a proof of A and a proof of B, so one may see such a type written as $A \wedge B$.

Σ-types are more powerful than typical ordered pair types because of dependent typing[25]. In the ordered pair, the type of the second term can depend on the value of the first term.

17 https://en.wikipedia.org/wiki/First-order_logic
18 https://en.wikipedia.org/wiki/Type_theory
19 https://en.wikipedia.org/wiki/Empty_type
20 https://en.wikipedia.org/wiki/Negation
21 https://en.wikipedia.org/wiki/Unit_type
22 https://en.wikipedia.org/wiki/Boolean_Algebra
23 https://en.wikipedia.org/wiki/Law_of_excluded_middle
24 https://en.wikipedia.org/wiki/Cartesian_product
25 https://en.wikipedia.org/wiki/Dependent_type

For example, the first term of the pair might be a natural number and the second term's type might be a vector of length equal to the first term. Such a type would be written:

$$\sum_{n:\mathbb{N}} \mathrm{Vec}(\mathbb{R}, n)$$

Using set-theory terminology, this is similar to an indexed disjoint unions[26] of sets. In the case of usual ordered pairs, the type of the second term does not depend on the value of the first term. Thus the type describing the cartesian product $\mathbb{N} \times \mathbb{R}$ is written:

$$\sum_{n:\mathbb{N}} \mathbb{R}$$

It is important to note here that the value of the first term, n, is not depended on by the type of the second term, $\mathbb{R}$.

Σ-types can be used to build up longer dependently-typed tuples[27] used in mathematics and the records or structs[28] used in most programming languages. An example of a dependently-typed 3-tuple is two integers and a proof that the first integer is smaller than the second integer, described by the type:

$$\sum_{m:\mathbb{Z}} \sum_{n:\mathbb{Z}} ((m < n) = \mathrm{True})$$

Dependent typing allows Σ-types to serve the role of existential quantifier[29]. The statement "there exists an n of type $\mathbb{N}$, such that $P(n)$ is proven" becomes the type of ordered pairs where the first item is the value n of type $\mathbb{N}$ and the second item is a proof of $P(n)$. Notice that the type of the second item (proofs of $P(n)$) depends on the value in the first part of the ordered pair (n). Its type would be:

$$\sum_{n:\mathbb{N}} P(n)$$

1.2.3 Π type constructor

Π-types contain functions. As with typical function types, they consist of an input type and an output type. They are more powerful than typical function types however, in that the return type can depend on the input value. Functions in type theory are different from set theory. In set theory, you look up the argument's value in a set of ordered pairs. In type theory, the argument is substituted into a term and then computation ("reduction") is applied to the term.

As an example, the type of a function that, given a natural number n, returns a vector containing n real numbers is written:

$$\prod_{n:\mathbb{N}} \mathrm{Vec}(\mathbb{R}, n)$$

When the output type does not depend on the input value, the function type is often simply written with a $\rightarrow$. Thus, $\mathbb{N} \rightarrow \mathbb{R}$ is the type of functions from natural numbers

26 https://en.wikipedia.org/wiki/Disjoint_union
27 https://en.wikipedia.org/wiki/Tuple
28 https://en.wikipedia.org/wiki/Record_(computer_science)
29 https://en.wikipedia.org/wiki/Existential_quantifier

to real numbers. Such Π-types correspond to logical implication. The logical proposition $A \Longrightarrow B$ corresponds to the type $A \to B$, containing functions that take proofs-of-A and return proofs-of-B. This type could be written more consistently as:

$$\prod_{a:A} B$$

Π-types are also used in logic for universal quantification[30]. The statement "for every n of type $\mathbb{N}$, $P(n)$ is proven" becomes a function from n of type $\mathbb{N}$ to proofs of $P(n)$. Thus, given the value for n the function generates a proof that $P()$ holds for that value. The type would be

$$\prod_{n:\mathbb{N}} P(n)$$

1.2.4 = type constructor

=-types are created from two terms. Given two terms like $2+2$ and $2 \cdot 2$, you can create a new type $2+2=2 \cdot 2$. The terms of that new type represent proofs that the pair reduce to the same canonical term. Thus, since both $2+2$ and $2 \cdot 2$ compute to the canonical term 4, there will be a term of the type $2+2=2 \cdot 2$. In intuitionistic type theory, there is a single way to make terms of =-types and that is by reflexivity[31]:

$$\text{refl} : \prod_{a:A} (a = a).$$

It is possible to create =-types such as $1=2$ where the terms do not reduce to the same canonical term, but you will be unable to create terms of that new type. In fact, if you were able to create a term of $1=2$, you could create a term of $\bot$. Putting that into a function would generate a function of type $1=2 \to \bot$. Since $\ldots \to \bot$ is how intuitionistic type theory defines negation, you would have $\neg(1=2)$ or, finally, $1 \neq 2$.

Equality of proofs is an area of active research in proof theory[32] and has led to the development of homotopy type theory[33] and other type theories.

1.2.5 Inductive types

Main article: Inductive type[34] Inductive types allow the creation of complex, self-referential types. For example, a linked list of natural numbers is either an empty list or a pair of a natural number and another linked list. Inductive types can be used to define unbounded mathematical structures like trees, graphs, etc.. In fact, the natural numbers type may be defined as an inductive type, either being 0 or the successor[35] of another natural number.

Inductive types define new constants, such as zero $0{:}\mathbb{N}$ and the successor function $S{:}\mathbb{N} \to \mathbb{N}$. Since S does not have a definition and cannot be evaluated using substitution, terms like $S0$ and $SSS0$ become the canonical terms of the natural numbers.

30 https://en.wikipedia.org/wiki/Universal_quantification
31 https://en.wikipedia.org/wiki/Reflexive_relation
32 https://en.wikipedia.org/wiki/Proof_theory
33 https://en.wikipedia.org/wiki/Homotopy_type_theory
34 https://en.wikipedia.org/wiki/Inductive_type
35 https://en.wikipedia.org/wiki/Successor_function

Proofs on inductive types are made possible by induction[36]. Each new inductive type comes with its own inductive rule. To prove a predicate $P()$ for every natural number, you use the following rule:

$$\mathbb{N}\text{-elim}: P(0) \to \left(\prod_{n:\mathbb{N}} P(n) \to P(S(n))\right) \to \prod_{n:\mathbb{N}} P(n)$$

Inductive types in intuitionistic type theory are defined in terms of W-types, the type of well-founded[37] trees. Later work in type theory generated coinductive types, induction-recursion, and induction-induction for working on types with more obscure kinds of self-referentiality. Higher inductive types[38] allow equality to be defined between terms.

1.2.6 Universe types

The universe types allow proofs to be written about all the types created with the other type constructors. Every term in the universe type $\mathcal{U}_0$ can be mapped to a type created with any combination of $0, 1, 2, \Sigma, \Pi, =$, and the inductive type constructor. However, to avoid paradoxes, there is no term in $\mathcal{U}_0$ that maps to $\mathcal{U}_0$.

To write proofs about all "the small types" and $\mathcal{U}_0$, you must use $\mathcal{U}_1$, which does contain a term for $\mathcal{U}_0$, but not for itself $\mathcal{U}_1$. Similarly, for $\mathcal{U}_2$. There is a predicative[39] hierarchy of universes, so to quantify a proof over any fixed constant k universes, you can use $\mathcal{U}_{k+1}$.

Universe types are a tricky feature of type theories. Martin-Löf's original type theory had to be changed to account for Girard's paradox[40]. Later research covered topics such as "super universes", "Mahlo universes", and impredicative universes.

1.3 Judgements

The formal definition of intuitionistic type theory is written using judgements. For example, in the statement "if A is a type and B is a type then $\sum_{a:A} B$ is a type" there are judgements of "is a type", "and", and "if ... then ...". The expression $\sum_{a:A} B$ is not a judgement; it is the type being defined.

This second level of the type theory can be confusing, particularly where it comes to equality. There is a judgement of term equality, which might say $4 = 2 + 2$. It is a statement that two terms reduce to the same canonical term. There is also a judgement of type equality, say that $A = B$, which means every element of A is an element of the type B and vice versa. At the type level, there is a type $4 = 2 + 2$ and it contains terms if there is a proof that 4 and $2 + 2$ reduce to the same value. (Terms of this type are generated using the term-equality judgement.) Lastly, there is an English-language level of equality, because we use the word "four" and symbol "4" to refer to the canonical term $SSSS0$. Synonyms like these are called "definitionally equal" by Martin-Löf.

36 https://en.wikipedia.org/wiki/Structural_induction
37 https://en.wikipedia.org/wiki/Well-founded
38 https://en.wikipedia.org/wiki/Higher_inductive_type
39 https://en.wikipedia.org/wiki/Impredicativity
40 https://en.wikipedia.org/wiki/Girard%27s_paradox

The description of judgements below is based on the discussion in Nordström, Petersson, and Smith.

The formal theory works with *types* and *objects*.

A type is declared by:

- A Type

An object exists and is in a type if:

- $a{:}A$

Objects can be equal

- $a = b$

and types can be equal

- $A = B$

A type that depends on an object from another type is declared

- $(x{:}A)B$

and removed by substitution

- $B[x/a]$, replacing the variable x with the object a in B.

An object that depends on an object from another type can be done two ways. If the object is "abstracted", then it is written

- $[x]b$

and removed by substitution

- $b[x/a]$, replacing the variable x with the object a in b.

The object-depending-on-object can also be declared as a constant as part of a recursive type. An example of a recursive type is:

- $0{:}\mathbb{N}$
- $S{:}\mathbb{N} \to \mathbb{N}$

Here, S is a constant object-depending-on-object. It is not associated with an abstraction. Constants like S can be removed by defining equality. Here the relationship with addition is defined using equality and using pattern matching to handle the recursive aspect of S:

$$\begin{aligned} \text{add: } (\mathbb{N} \times \mathbb{N}) &\to \mathbb{N} \\ \text{add}(0, b) &= b \\ \text{add}(S(a), b) &= S(\text{add}(a, b))) \end{aligned}$$

S is manipulated as an opaque constant - it has no internal structure for substitution.

So, objects and types and these relations are used to express formulae in the theory. The following styles of judgements are used to create new objects, types and relations from existing ones:

$\Gamma \vdash \sigma$ Type	σ is a well-formed type in the context Γ.

$\Gamma \vdash t{:}\sigma$	t is a well-formed term of type σ in context Γ.
$\Gamma \vdash \sigma \equiv \tau$	σ and τ are equal types in context Γ.
$\Gamma \vdash t \equiv u{:}\sigma$	t and u are judgmentally equal terms of type σ in context Γ.
$\vdash \Gamma$ Context	Γ is a well-formed context of typing assumptions.

By convention, there is a type that represents all other types. It is called $\mathcal{U}$ (or Set). Since $\mathcal{U}$ is a type, the members of it are objects. There is a dependent type El that maps each object to its corresponding type. *In most texts* El *is never written.* From the context of the statement, a reader can almost always tell whether A refers to a type, or whether it refers to the object in $\mathcal{U}$ that corresponds to the type.

This is the complete foundation of the theory. Everything else is derived.

To implement logic, each proposition is given its own type. The objects in those types represent the different possible ways to prove the proposition. If there is no proof for the proposition, then the type has no objects in it. Operators like "and" and "or" that work on propositions introduce new types and new objects. So $A \times B$ is a type that depends on the type A and the type B. The objects in that dependent type are defined to exist for every pair of objects in A and B. If A or B has no proof and is an empty type, then the new type representing $A \times B$ is also empty.

This can be done for other types (booleans, natural numbers, etc.) and their operators.

1.4 Categorical models of type theory

Using the language of category theory[41], R. A. G. Seely[42] introduced the notion of a locally cartesian closed category[43] (LCCC) as the basic model of type theory. This has been refined by Hofmann and Dybjer to *Categories with Families* or *Categories with Attributes* based on earlier work by Cartmell.[1]

A category with families is a category C of contexts (in which the objects are contexts, and the context morphisms are substitutions), together with a functor $T: C^{\mathrm{op}} \to \mathit{Fam}(\mathit{Set})$.

$\mathit{Fam}(\mathit{Set})$ is the category of families[44] of Sets, in which objects are pairs (A, B) of an "index set" A and a function B: $X \to A$, and morphisms are pairs of functions $f: A \to A'$ and $g: X \to X'$, such that $B' \circ g = f \circ B$ — in other words, f maps B_{a} to $B_{g(a)}$.

The functor T assigns to a context G a set $Ty(G)$ of types, and for each $A : Ty(G)$, a set $Tm(G, A)$ of terms. The axioms for a functor require that these play harmoniously with substitution. Substitution is usually written in the form Af or af, where A is a type in $Ty(G)$ and a is a term in $Tm(G, A)$, and f is a substitution from D to G. Here $Af : Ty(D)$ and $af : Tm(D, Af)$.

The category C must contain a terminal object (the empty context), and a final object for a form of product called comprehension, or context extension, in which the right element is

41 https://en.wikipedia.org/wiki/Category_theory
42 https://en.wikipedia.org/w/index.php?title=R._A._G._Seely&action=edit&redlink=1
43 https://en.wikipedia.org/wiki/Locally_cartesian_closed_category
44 https://en.wikipedia.org/w/index.php?title=Category_of_families&action=edit&redlink=1

a type in the context of the left element. If G is a context, and $A : Ty(G)$, then there should be an object (G, A) final among contexts D with mappings $p : D \to G$, $q : Tm(D,Ap)$.

A logical framework, such as Martin-Löf's takes the form of closure conditions on the context dependent sets of types and terms: that there should be a type called Set, and for each set a type, that the types should be closed under forms of dependent sum and product, and so forth.

A theory such as that of predicative set theory expresses closure conditions on the types of sets and their elements: that they should be closed under operations that reflect dependent sum and product, and under various forms of inductive definition.

1.5 Extensional versus intensional

A fundamental distinction is extensional[45] vs intensional[46] type theory. In extensional type theory definitional (i.e., computational) equality is not distinguished from propositional equality, which requires proof. As a consequence type checking becomes undecidable[47] in extensional type theory because programs in the theory might not terminate. For example, such a theory allows one to give a type to the Y-combinator[48], a detailed example of this can be found in Nordstöm and Petersson *Programming in Martin-Löf's Type Theory.*[2] However, this doesn't prevent extensional type theory from being a basis for a practical tool, for example, NuPRL[49] is based on extensional type theory.

In contrast in intensional type theory type checking[50] is decidable[51], but the representation of standard mathematical concepts is somewhat more cumbersome, since intensional reasoning requires using setoids[52] or similar constructions. There are many common mathematical objects, which are hard to work with or can't be represented without this, for example, integer numbers[53], rational numbers[54], and real numbers[55]. Integers and rational numbers can be represented without setoids, but this representation isn't easy to work with. Cauchy real numbers can't be represented without this.[3]

Homotopy type theory[56] works on resolving this problem. It allows one to define higher inductive types[57], which not only define first order constructors (values[58] or points[59]), but

45 https://en.wikipedia.org/wiki/Extensional_definition
46 https://en.wikipedia.org/wiki/Intensional_definition
47 https://en.wikipedia.org/wiki/Undecidable_problem
48 https://en.wikipedia.org/wiki/Fixpoint_combinator
49 https://en.wikipedia.org/wiki/NuPRL
50 https://en.wikipedia.org/wiki/Type_checking
51 https://en.wikipedia.org/wiki/Decision_problem
52 https://en.wikipedia.org/wiki/Setoid
53 https://en.wikipedia.org/wiki/Integer
54 https://en.wikipedia.org/wiki/Rational_number
55 https://en.wikipedia.org/wiki/Real_number
56 https://en.wikipedia.org/wiki/Homotopy_type_theory
57 https://en.wikipedia.org/wiki/Higher_inductive_type
58 https://en.wikipedia.org/wiki/Value_(computer_science)
59 https://en.wikipedia.org/wiki/Point_(geometry)

higher order constructors, i.e. equalities between elements (paths[60]), equalities between equalities (homotopies[61]), *ad infinitum*.

1.6 Implementations of type theory

Different forms of type theory have been implemented as the formal systems underlying of a number of proof assistants[62]. While many are based on Per Martin-Löf's ideas, many have added features, more axioms, or different philosophical background. For instance, the NuPRL[63] system is based on computational type theory[64][4] and Coq[65] is based on the calculus of (co)inductive constructions[66]. Dependent types[67] also feature in the design of programming languages[68] such as ATS[69], Cayenne[70], Epigram[71], Agda[72],[5] and Idris[73].[6]

1.7 Martin-Löf type theories

Per Martin-Löf[74] constructed several type theories that were published at various times, some of them much later than when the preprints with their description became accessible to the specialists (among others Jean-Yves Girard[75] and Giovanni Sambin). The list below attempts to list all the theories that have been described in a printed form and to sketch the key features that distinguished them from each other. All of these theories had dependent products, dependent sums, disjoint unions, finite types and natural numbers. All the theories had the same reduction rules that did not include η-reduction either for dependent products or for dependent sums, except for MLTT79 where the η-reduction for dependent products is added.

MLTT71 was the first of type theories created by Per Martin-Löf. It appeared in a preprint in 1971. It had one universe but this universe had a name in itself, i.e. it was a type theory with, as it is called today, "Type in Type". Jean-Yves Girard[76] has shown that this system was inconsistent and the preprint was never published.

MLTT72 was presented in a 1972 preprint that has now been published.[7] That theory had one universe V and no identity types[*definition needed*[77]]. The universe was "predicative" in

60 https://en.wikipedia.org/wiki/Path_(topology)
61 https://en.wikipedia.org/wiki/Homotopy
62 https://en.wikipedia.org/wiki/Proof_assistant
63 https://en.wikipedia.org/wiki/NuPRL
64 https://en.wikipedia.org/w/index.php?title=Computational_type_theory&action=edit&redlink=1
65 https://en.wikipedia.org/wiki/Coq
66 https://en.wikipedia.org/wiki/Calculus_of_constructions
67 https://en.wikipedia.org/wiki/Dependent_types
68 https://en.wikipedia.org/wiki/Programming_languages
69 https://en.wikipedia.org/wiki/ATS_(programming_language)
70 https://en.wikipedia.org/wiki/Cayenne_(programming_language)
71 https://en.wikipedia.org/wiki/Epigram_(programming_language)
72 https://en.wikipedia.org/wiki/Agda_(theorem_prover)
73 https://en.wikipedia.org/wiki/Idris_(programming_language)
74 https://en.wikipedia.org/wiki/Per_Martin-L%C3%B6f
75 https://en.wikipedia.org/wiki/Jean-Yves_Girard
76 https://en.wikipedia.org/wiki/Jean-Yves_Girard

the sense that the dependent product of a family of objects from V over an object that was not in V such as, for example, V itself, was not assumed to be in V. The universe was à la Russell, i.e., one would write directly "T∈V" and "t∈T" (Martin-Löf uses the sign "∈" instead of modern ":") without the additional constructor such as "El".

MLTT73 was the first definition of a type theory that Per Martin-Löf published (it was presented at the Logic Colloquium 73 and published in 1975[8]). There are identity types which he calls "propositions" but since no real distinction between propositions and the rest of the types is introduced the meaning of this is unclear. There is what later acquires the name of J-eliminator but yet without a name (see pp. 94–95). There is in this theory an infinite sequence of universes V_0, ..., V_n, ... The universes are predicative, a-la Russell and *non-cumulative*! In fact, Corollary 3.10 on p. 115 says that if A∈V_m and B∈V_n are such that A and B are convertible then $m = n$. This means, for example, that it would be difficult to formulate univalence in this theory—there are contractible types in each of the V_i but it is unclear how to declare them to be equal since there are no identity types connecting V_i and V_j for $i \neq j$.

MLTT79 was presented in 1979 and published in 1982.[9] In this paper, Martin-Löf introduced the four basic types of judgement for the dependent type theory that has since become fundamental in the study of the meta-theory of such systems. He also introduced contexts as a separate concept in it (see p. 161). There are identity types with the J-eliminator (which already appeared in MLTT73 but did not have this name there) but also with the rule that makes the theory "extensional" (p. 169). There are W-types. There is an infinite sequence of predicative universes that *are cumulative*.

Bibliopolis: there is a discussion of a type theory in the Bibliopolis book from 1984[10] but it is somewhat open-ended and does not seem to represent a particular set of choices and so there is no specific type theory associated with it.

1.8 See also

- Intuitionistic logic[78]
- Typed lambda calculus[79]

1.9 Notes

1. Clairambault, Pierre; Dybjer, Peter (2014). "The biequivalence of locally cartesian closed categories and Martin-Löf type theories"[80].

78 https://en.wikipedia.org/wiki/Intuitionistic_logic

79 https://en.wikipedia.org/wiki/Typed_lambda_calculus

80 https://www.cambridge.org/core/journals/mathematical-structures-in-computer-science/article/biequivalence-of-locally-cartesian-closed-categories-and-martinlof-type-theories/6ECB295B1246A85D5DD92E5F38428D99

Mathematical Structures in Computer Science. **24** (6). arXiv[81]:1112.3456[82]. doi[83]:10.1017/S0960129513000881[84]. ISSN[85] 0960-1295[86].

2. Bengt Nordström; Kent Petersson; Jan M. Smith (1990). *Programming in Martin-Löf's Type Theory.* Oxford University Press, p. 90.
3. Altenkirch, Thorsten, Thomas Anberrée, and Nuo Li. "Definable Quotients in Type Theory."
4. Allen, S.F.; Bickford, M.; Constable, R.L.; Eaton, R.; Kreitz, C.; Lorigo, L.; Moran, E. (2006). "Innovations in computational type theory using Nuprl"[87]. *Journal of Applied Logic.* **4** (4): 428–469. doi[88]:10.1016/j.jal.2005.10.005[89].
5. Norell, Ulf (2009). *Dependently Typed Programming in Agda. Proceedings of the 4th International Workshop on Types in Language Design and Implementation* . TLDI '09. New York, NY, USA: ACM. pp. 1–2. CiteSeerX[90] 10.1.1.163.7149[91]. doi[92]:10.1145/1481861.1481862[93]. ISBN[94] 9781605584201[95].
6. Brady, Edwin (2013). "Idris, a general-purpose dependently typed programming language: Design and implementation"[96]. *Journal of Functional Programming.* **23** (5): 552–593. doi[97]:10.1017/S095679681300018X[98]. ISSN[99] 0956-7968[100].
7. Per Martin-Löf, An intuitionistic theory of types, Twenty-five years of constructive type theory (Venice,1995), Oxford Logic Guides, v. 36, pp. 127--172, Oxford Univ. Press, New York, 1998
8. Per Martin-Löf, An intuitionistic theory of types: predicative part, Logic Colloquium '73 (Bristol, 1973), 73--118. Studies in Logic and the Foundations of Mathematics, Vol. 80, North-Holland, Amsterdam,1975
9. Per Martin-Löf, Constructive mathematics and computer programming, Logic, methodology and philosophy of science, VI (Hannover, 1979), Stud. Logic Found. Math., v. 104, pp. 153--175, North-Holland, Amsterdam, 1982
10. Per Martin-Löf, Intuitionistic type theory, Studies in Proof Theory. Lecture Notes, v. 1, Notes by Giovanni Sambin, pp. iv+91, 1984

81 https://en.wikipedia.org/wiki/ArXiv_(identifier)
82 http://arxiv.org/abs/1112.3456
83 https://en.wikipedia.org/wiki/Doi_(identifier)
84 https://doi.org/10.1017%2FS0960129513000881
85 https://en.wikipedia.org/wiki/ISSN_(identifier)
86 http://www.worldcat.org/issn/0960-1295
87 https://doi.org/10.1016%2Fj.jal.2005.10.005
88 https://en.wikipedia.org/wiki/Doi_(identifier)
89 https://doi.org/10.1016%2Fj.jal.2005.10.005
90 https://en.wikipedia.org/wiki/CiteSeerX_(identifier)
91 http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.163.7149
92 https://en.wikipedia.org/wiki/Doi_(identifier)
93 https://doi.org/10.1145%2F1481861.1481862
94 https://en.wikipedia.org/wiki/ISBN_(identifier)
95 https://en.wikipedia.org/wiki/Special:BookSources/9781605584201
https://www.cambridge.org/core/journals/journal-of-functional-programming/
96 article/idris-a-generalpurpose-dependently-typed-programming-language-design-and-implementation/418409138B4452969AC0736DB0A2C238
97 https://en.wikipedia.org/wiki/Doi_(identifier)
98 https://doi.org/10.1017%2FS095679681300018X
99 https://en.wikipedia.org/wiki/ISSN_(identifier)
100 http://www.worldcat.org/issn/0956-7968

1.10 References

- Martin-Löf, Per (1984). *Intuitionistic type theory*[101] (PDF). Sambin, Giovanni. Napoli: Bibliopolis. ISBN[102] 978-8870881059[103]. OCLC[104] 12731401[105].

1.11 Further reading

- Per Martin-Löf's Notes, as recorded by Giovanni Sambin (1980)[106]
- Nordström, Bengt; Petersson, Kent; Smith, Jan M. (1990). *Programming in Martin-Löf's Type Theory*[107]. Oxford University Press. ISBN[108] 9780198538141[109].
- Thompson, Simon (1991). *Type Theory and Functional Programming*[110]. Addison-Wesley. ISBN[111] 0-201-41667-0[112].
- Granström, Johan G. (2011). *Treatise on Intuitionistic Type Theory*[113]. Springer. ISBN[114] 978-94-007-1735-0[115].

1.12 External links

- EU Types Project: Tutorials[116] – lecture notes and slides from the Types Summer School 2005
- n-Categories - Sketch of a Definition[117] – letter from John Baez[118] and James Dolan to Ross Street[119], November 29, 1995

Non-classical logic

101 http://intuitionistic.files.wordpress.com/2010/07/martin-lof-tt.pdf
102 https://en.wikipedia.org/wiki/ISBN_(identifier)
103 https://en.wikipedia.org/wiki/Special:BookSources/978-8870881059
104 https://en.wikipedia.org/wiki/OCLC_(identifier)
105 http://www.worldcat.org/oclc/12731401
106 http://www.cse.chalmers.se/~peterd/papers/MartinL%C3%B6f1984.pdf
107 http://www.cs.chalmers.se/Cs/Research/Logic/book/
108 https://en.wikipedia.org/wiki/ISBN_(identifier)
109 https://en.wikipedia.org/wiki/Special:BookSources/9780198538141
110 http://www.cs.kent.ac.uk/people/staff/sjt/TTFP/
111 https://en.wikipedia.org/wiki/ISBN_(identifier)
112 https://en.wikipedia.org/wiki/Special:BookSources/0-201-41667-0
113 https://www.springer.com/philosophy/book/978-94-007-1735-0
114 https://en.wikipedia.org/wiki/ISBN_(identifier)
115 https://en.wikipedia.org/wiki/Special:BookSources/978-94-007-1735-0
116 http://www.cs.chalmers.se/Cs/Research/Logic/Types/tutorials.html
117 http://math.ucr.edu/home/baez/ncat.def.html
118 https://en.wikipedia.org/wiki/John_Baez
119 https://en.wikipedia.org/wiki/Ross_Street

Major topics in Foundations of Mathematics

- This page was last edited on 22 January 2022, at 08:11 (UTC).
- Text is available under the Creative Commons Attribution-ShareAlike License 3.0[120][121]; additional terms may apply. By using this site, you agree to the Terms of Use[122] and Privacy Policy[123]. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc.[124], a non-profit organization.

120 http://en.wikipedia.org/wiki/Wikipedia:Text_of_Creative_Commons_Attribution-ShareAlike_3.0_Unported_License
121 http://creativecommons.org/licenses/by-sa/3.0/
122 http://foundation.wikimedia.org/wiki/Terms_of_Use
123 http://foundation.wikimedia.org/wiki/Privacy_policy
124 http://www.wikimediafoundation.org/

2 Contributors

Edits	User
1	Adavidb[1]
8	Ancheta Wis[2]
2	Arademaker[3]
4	AxelBoldt[4]
1	BD2412[5]
1	BG19bot[6]
2	Bbbaat[7]
2	Ben Standeven[8]
1	BiT[9]
1	Brad7777[10]
1	Brighterorange[11]
4	Cedar101[12]
1	Citation bot[13]
1	Classicalecon[14]
1	Cobi[15]
1	Codingmasters[16]
2	Cyberbot II[17]
2	Cydebot[18]
2	Dagit[19]
1	Daira Hopwood[20]
1	Daniel5Ko[21]

1 https://en.wikipedia.org/wiki/User:Adavidb
2 https://en.wikipedia.org/wiki/User:Ancheta_Wis
3 https://en.wikipedia.org/wiki/User:Arademaker
4 https://en.wikipedia.org/wiki/User:AxelBoldt
5 https://en.wikipedia.org/wiki/User:BD2412
6 https://en.wikipedia.org/wiki/User:BG19bot
7 https://en.wikipedia.org/w/index.php%3ftitle=User:Bbbaat&action=edit&redlink=1
8 https://en.wikipedia.org/wiki/User:Ben_Standeven
9 https://en.wikipedia.org/wiki/User:BiT
10 https://en.wikipedia.org/wiki/User:Brad7777
11 https://en.wikipedia.org/wiki/User:Brighterorange
12 https://en.wikipedia.org/w/index.php%3ftitle=User:Cedar101&action=edit&redlink=1
13 https://en.wikipedia.org/wiki/User:Citation_bot
14 https://en.wikipedia.org/wiki/User:Classicalecon
15 https://en.wikipedia.org/wiki/User:Cobi
16 https://en.wikipedia.org/wiki/User:Codingmasters
17 https://en.wikipedia.org/wiki/User:Cyberbot_II
18 https://en.wikipedia.org/wiki/User:Cydebot
19 https://en.wikipedia.org/w/index.php%3ftitle=User:Dagit&action=edit&redlink=1
20 https://en.wikipedia.org/wiki/User:Daira_Hopwood
21 https://en.wikipedia.org/w/index.php%3ftitle=User:Daniel5Ko&action=edit&redlink=1

1 Diego Moya[22]
1 Donner60[23]
1 Dunham[24]
1 Elwikipedista~enwiki[25]
1 EmilJ[26]
1 Equinox[27]
1 Fblanqui[28]
8 FeliksK[29]
1 Franka W[30]
1 FrescoBot[31]
1 Fropuff[32]
1 Functor salad[33]
1 Fylwind[34]
1 Genneth[35]
1 Giftlite[36]
2 Greenrd[37]
4 Gregbard[38]
1 GünniX[39]
9 Hairy Dude[40]
1 Harryboyles[41]
3 Helpful Pixie Bot[42]
1 Hiiiiiiiiiiiiiiiiiii[43]
1 Hjfreyer[44]
1 Hyacinth[45]
1 Ianushii[46]

22 https://en.wikipedia.org/wiki/User:Diego_Moya
23 https://en.wikipedia.org/wiki/User:Donner60
24 https://en.wikipedia.org/wiki/User:Dunham
25 https://en.wikipedia.org/wiki/User:Elwikipedista~enwiki
26 https://en.wikipedia.org/wiki/User:EmilJ
27 https://en.wikipedia.org/wiki/User:Equinox
28 https://en.wikipedia.org/wiki/User:Fblanqui
29 https://en.wikipedia.org/w/index.php%3ftitle=User:FeliksK&action=edit&redlink=1
30 https://en.wikipedia.org/wiki/User:Franka_W
31 https://en.wikipedia.org/wiki/User:FrescoBot
32 https://en.wikipedia.org/wiki/User:Fropuff
33 https://en.wikipedia.org/wiki/User:Functor_salad
34 https://en.wikipedia.org/wiki/User:Fylwind
35 https://en.wikipedia.org/wiki/User:Genneth
36 https://en.wikipedia.org/wiki/User:Giftlite
37 https://en.wikipedia.org/wiki/User:Greenrd
38 https://en.wikipedia.org/wiki/User:Gregbard
39 https://en.wikipedia.org/wiki/User:G%25C3%25BCnniX
40 https://en.wikipedia.org/wiki/User:Hairy_Dude
41 https://en.wikipedia.org/wiki/User:Harryboyles
42 https://en.wikipedia.org/wiki/User:Helpful_Pixie_Bot
43 https://en.wikipedia.org/w/index.php%3ftitle=User:Hiiiiiiiiiiiiiiiiiii&action=edit&redlink=1
44 https://en.wikipedia.org/wiki/User:Hjfreyer
45 https://en.wikipedia.org/wiki/User:Hyacinth
46 https://en.wikipedia.org/w/index.php%3ftitle=User:Ianushii&action=edit&redlink=1

1 Ibic[47]
1 Jarble[48]
1 Jbergquist[49]
1 Jim Apple[50]
4 Jkliff[51]
2 Jochen Burghardt[52]
1 Joel Brennan[53]
1 JohnBlackburne[54]
4 Jon Awbrey[55]
2 JonathanHopeThisIsUnique[56]
2 KartikSinghal[57]
1 Khazar2[58]
1 Kimbly[59]
1 Kirelagin[60]
1 KolbertBot[61]
4 Konstantin.Solomatov[62]
1 Lambiam[63]
1 Legobot[64]
1 Lightbot[65]
1 LilHelpa[66]
1 Magic links bot[67]
2 MagnusFit[68]
2 Matěj Grabovský[69]
8 Maxdamantus[70]
33 Mdnahas[71]

47 https://en.wikipedia.org/w/index.php%3ftitle=User:Ibic&action=edit&redlink=1
48 https://en.wikipedia.org/wiki/User:Jarble
49 https://en.wikipedia.org/wiki/User:Jbergquist
50 https://en.wikipedia.org/w/index.php%3ftitle=User:Jim_Apple&action=edit&redlink=1
51 https://en.wikipedia.org/w/index.php%3ftitle=User:Jkliff&action=edit&redlink=1
52 https://en.wikipedia.org/wiki/User:Jochen_Burghardt
53 https://en.wikipedia.org/w/index.php%3ftitle=User:Joel_Brennan&action=edit&redlink=1
54 https://en.wikipedia.org/wiki/User:JohnBlackburne
55 https://en.wikipedia.org/wiki/User:Jon_Awbrey
56 https://en.wikipedia.org/wiki/User:JonathanHopeThisIsUnique
57 https://en.wikipedia.org/w/index.php%3ftitle=User:KartikSinghal&action=edit&redlink=1
58 https://en.wikipedia.org/wiki/User:Khazar2
59 https://en.wikipedia.org/wiki/User:Kimbly
60 https://en.wikipedia.org/wiki/User:Kirelagin
61 https://en.wikipedia.org/wiki/User:KolbertBot
62 https://en.wikipedia.org/wiki/User:Konstantin.Solomatov
63 https://en.wikipedia.org/wiki/User:Lambiam
64 https://en.wikipedia.org/wiki/User:Legobot
65 https://en.wikipedia.org/wiki/User:Lightbot
66 https://en.wikipedia.org/wiki/User:LilHelpa
67 https://en.wikipedia.org/wiki/User:Magic_links_bot
68 https://en.wikipedia.org/w/index.php%3ftitle=User:MagnusFit&action=edit&redlink=1
69 https://en.wikipedia.org/wiki/User:Mat%25C4%259Bj_Grabovsk%25C3%25BD
70 https://en.wikipedia.org/wiki/User:Maxdamantus
71 https://en.wikipedia.org/wiki/User:Mdnahas

1 Me, Myself, and I are Here[72]
1 Mhss[73]
6 Michael Hardy[74]
1 Mikon[75]
1 Monkbot[76]
1 Noamz[77]
2 OAbot[78]
3 Oleg Alexandrov[79]
3 Omnipaedista[80]
1 PWilkinson[81]
1 Patrikj[82]
3 Pcap[83]
1 Pmetzger[84]
1 PythonGraham[85]
1 R'n'B[86]
1 RDBrown[87]
1 Rcog[88]
2 Rich Farmbrough[89]
3 Ruud Koot[90]
1 STyx[91]
1 Sagaciousuk[92]
1 Sam Staton[93]
1 Sankura[94]
1 ShelfSkewed[95]
5 Siddharthist[96]

72 https://en.wikipedia.org/wiki/User:Me,_Myself,_and_I_are_Here
73 https://en.wikipedia.org/wiki/User:Mhss
74 https://en.wikipedia.org/wiki/User:Michael_Hardy
75 https://en.wikipedia.org/w/index.php%3ftitle=User:Mikon&action=edit&redlink=1
76 https://en.wikipedia.org/wiki/User:Monkbot
77 https://en.wikipedia.org/wiki/User:Noamz
78 https://en.wikipedia.org/wiki/User:OAbot
79 https://en.wikipedia.org/wiki/User:Oleg_Alexandrov
80 https://en.wikipedia.org/wiki/User:Omnipaedista
81 https://en.wikipedia.org/wiki/User:PWilkinson
82 https://en.wikipedia.org/wiki/User:Patrikj
83 https://en.wikipedia.org/wiki/User:Pcap
84 https://en.wikipedia.org/wiki/User:Pmetzger
85 https://en.wikipedia.org/wiki/User:PythonGraham
86 https://en.wikipedia.org/wiki/User:R%2527n%2527B
87 https://en.wikipedia.org/wiki/User:RDBrown
88 https://en.wikipedia.org/wiki/User:Rcog
89 https://en.wikipedia.org/wiki/User:Rich_Farmbrough
90 https://en.wikipedia.org/wiki/User:Ruud_Koot
91 https://en.wikipedia.org/wiki/User:STyx
92 https://en.wikipedia.org/wiki/User:Sagaciousuk
93 https://en.wikipedia.org/wiki/User:Sam_Staton
94 https://en.wikipedia.org/w/index.php%3ftitle=User:Sankura&action=edit&redlink=1
95 https://en.wikipedia.org/wiki/User:ShelfSkewed
96 https://en.wikipedia.org/wiki/User:Siddharthist

5 Tea2min[97]
1 The Eloquent Peasant[98]
2 Tijfo098[99]
2 Toby Bartels[100]
32 Txa[101]
1 Varkora[102]
1 Vegaswikian[103]
2 Vlad Patryshev[104]
1 Vladimirias[105]
2 Woohookitty[106]
3 XLinkBot[107]
1 Yobot[108]

97 https://en.wikipedia.org/wiki/User:Tea2min
98 https://en.wikipedia.org/wiki/User:The_Eloquent_Peasant
99 https://en.wikipedia.org/wiki/User:Tijfo098
100 https://en.wikipedia.org/wiki/User:Toby_Bartels
101 https://en.wikipedia.org/wiki/User:Txa
102 https://en.wikipedia.org/w/index.php%3ftitle=User:Varkora&action=edit&redlink=1
103 https://en.wikipedia.org/wiki/User:Vegaswikian
104 https://en.wikipedia.org/wiki/User:Vlad_Patryshev
105 https://en.wikipedia.org/wiki/User:Vladimirias
106 https://en.wikipedia.org/wiki/User:Woohookitty
107 https://en.wikipedia.org/wiki/User:XLinkBot
108 https://en.wikipedia.org/wiki/User:Yobot

List of Figures

- GFDL: Gnu Free Documentation License. http://www.gnu.org/licenses/fdl.html
- cc-by-sa-3.0: Creative Commons Attribution ShareAlike 3.0 License. http://creativecommons.org/licenses/by-sa/3.0/
- cc-by-sa-2.5: Creative Commons Attribution ShareAlike 2.5 License. http://creativecommons.org/licenses/by-sa/2.5/
- cc-by-sa-2.0: Creative Commons Attribution ShareAlike 2.0 License. http://creativecommons.org/licenses/by-sa/2.0/
- cc-by-sa-1.0: Creative Commons Attribution ShareAlike 1.0 License. http://creativecommons.org/licenses/by-sa/1.0/
- cc-by-2.0: Creative Commons Attribution 2.0 License. http://creativecommons.org/licenses/by/2.0/
- cc-by-2.0: Creative Commons Attribution 2.0 License. http://creativecommons.org/licenses/by/2.0/deed.en
- cc-by-2.5: Creative Commons Attribution 2.5 License. http://creativecommons.org/licenses/by/2.5/deed.en
- cc-by-3.0: Creative Commons Attribution 3.0 License. http://creativecommons.org/licenses/by/3.0/deed.en
- GPL: GNU General Public License. http://www.gnu.org/licenses/gpl-2.0.txt
- LGPL: GNU Lesser General Public License. http://www.gnu.org/licenses/lgpl.html
- PD: This image is in the public domain.
- ATTR: The copyright holder of this file allows anyone to use it for any purpose, provided that the copyright holder is properly attributed. Redistribution, derivative work, commercial use, and all other use is permitted.
- EURO: This is the common (reverse) face of a euro coin. The copyright on the design of the common face of the euro coins belongs to the European Commission. Authorised is reproduction in a format without relief (drawings, paintings, films) provided they are not detrimental to the image of the euro.
- LFK: Lizenz Freie Kunst. http://artlibre.org/licence/lal/de
- CFR: Copyright free use.

- EPL: Eclipse Public License. `http://www.eclipse.org/org/documents/epl-v10.php`

Copies of the GPL, the LGPL as well as a GFDL are included in chapter Licenses[109]. Please note that images in the public domain do not require attribution. You may click on the image numbers in the following table to open the webpage of the images in your webbrower.

109 Chapter 3 on page 27

3 Licenses

3.1 GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow. TERMS AND CONDITIONS 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion. 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work. 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary. 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures. 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee. 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

* a) The work must carry prominent notices stating that you modified it, and giving a relevant date. * b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices". * c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it. * d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate. 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

* a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange. * b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge. * c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b. * d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements. * e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying. 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

* a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or * b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or * c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or * d) Limiting the use for publicity purposes of names of licensors or authors of the material; or * e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or * f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way. 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10. 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so. 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it. 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law. 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy

both those terms and this License would be to refrain entirely from conveying the Program. 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such. 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public 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.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version. 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author> This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

3.2 GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. 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.

The "publisher" means any person or entity that distributes copies of the Document to the public.

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 under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it. 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. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document. 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing. 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 1.3 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.

3.3 GNU Lesser General Public License

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below. 0. Additional Definitions.

As used herein, “this License” refers to version 3 of the GNU Lesser General Public License, and the “GNU GPL” refers to version 3 of the GNU General Public License.

“The Library” refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An “Application” is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A “Combined Work” is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the “Linked Version”.

The “Minimal Corresponding Source” for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The “Corresponding Application Code” for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work. 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL. 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

* a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or * b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

* a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

* a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the Combined Work with a copy of the GNU GPL and this license document. * c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document. * d) Do one of the following: o 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source. o 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user’s computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version. * e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

* a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License. * b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public 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.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy’s public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.