MDPI

Article

Why Exp-Minus-Log (eml): towards an explanation based on optimization and symmetry

Vladik Kreinovich[1,‡], Leobardo Valera[2,‡], Igor Atamanyuk[3,4,‡*], and Yuriy Kondratenko[5,‡]

[1] Department of Computer Science, University of Texas at El Paso, 500 W. University, El Paso, Texas 79968, USA; vladik@utep.edu
[2] Computer Science and Information Technology Systems Department, El Paso Community College, 919 Hunter, El Paso, TX 79915, USA; leobardovalera@gmail.com
[3] Department of Applied Mathematics, Warsaw University of Life Sciences, Warsaw, Poland; ihor_atamaniuk@sggw.edu.pl
[4] Mykolaiv National Agrarian University, Mykolaiv, Ukraine
[5] Intelligent Information Systems Department, Petro Mohyla Black Sea National University, 10, 68th Desantnykiv Str., Mykolaiv 54003 Ukraine; y_kondrat2002@yahoo.com
* Correspondence: ihor_atamaniuk@sggw.edu.pl (I.A.)
‡ These authors contributed equally to this work.

Abstract: It has been recently shown that, similarly to the fact that every Boolean operation can be performed if we only have gates of a single type – NAND or NOR – many elementary functions can be obtained by using a constant 1 and a single function $\mathrm{eml}(x, y) = \exp(x) - \ln(y)$. This function is known as exp-minus-log, eml, for short. Just like the use of NAND gates led to computation speed-up, this new discovery has a potential for a similar speed-up. There are several other functions with the same property, but among them, eml is the fastest to compute. In this paper, we provide mathematical arguments that, in some reasonable sense, eml is probably optimal. In these arguments, we use a symmetry-based approach and its relation to optimization.

Keywords: eml function, exp-minus-log, optimization, symmetry, invariance

1. Formulation of the problem

In many practical situations, a universal tool is better than specialized ones. In general, when we repeatedly have to perform the same task, it is beneficial to design a specialized tool for this particular task. However, when we face a large variety of tasks, it is too complicated to design specialized tools for each of these tasks – especially in situations when new similar tasks appear all the time. In such situations, it is desirable to have a single universal tool that can handle all these tasks.

In particular, the use of universal tools is beneficial for computing. This is true in engineering, this is also true in computing. For example, Charles Babbage, one of the computing pioneers, first started with designing a specialized computational device – called *Difference Engine* – and then realized that it is beneficial to come up with a universal computational device – that he called *Analytical Engine*. Similarly, in the 20th century, first appeared specialized computational devices, and then it was realized that it is beneficial to come up with a universal device – a computer.

In designing a computer, researchers realized that it is often beneficial to minimize the number of supported operations – and these operations must be as simple as possible. Let us give some well-known examples about modern computers.

Received:
Revised:
Accepted:
Published:

Citation: Kreinovich, V.; Valera, L.; Atamanyuk, I.: Kondratenko, Yu. Why Exp-Minus-Log (eml): towards an explanation based on optimization and symmetry. *Mathematics* **2026**, *1*, 0. https://doi.org/

Copyright: © 2026 by the authors. Submitted to *Mathematics* for possible open access publication under the terms and conditions of the Creative Commons Attri- bution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

- We do not have a separately supported subtraction operation. Instead, we have addition $a + b$ and unary minus $-a$, so that $a - b$ is computed as $a + (-b)$.
- Instead of supporting division a/b, we use multiplication $a \cdot b$ and inverse $1/a$, so that a/b is computed as $a \cdot (1/b)$.
- Instead of supporting all possible Boolean operations, computers only support a few of them, and all other operations are computed as a combination of supported ones. For example, it is known that we can compute any Boolean expression if we have only one operation – this can be either $\mathrm{NAND}(x, y) \stackrel{\text{def}}{=} \neg(x \,\&\, y)$ or $\mathrm{NOR}(x, y) \stackrel{\text{def}}{=} \neg(x \vee y)$. This result is also used in the design of computer chips, where NAND gates are actively used.

We can come up with even more universal tools: a recent result. In spite of all the above reductions, modern computers still perform a large number of different operations – operations that are all supported either on the hardware level, or on the microprogramming level. Specifically, we have support for addition, for unary minus, for multiplication, for inverse, and for many special functions such as $\ln(x)$, $\exp(x)$, $\sin(x)$, etc.

A recent breakthrough result [7,8] has shown that all arithmetic operations (addition, subtraction, multiplication, division), several basic elementary functions ($\exp(x)$, $\ln(x)$, x^y, trigonometric functions, etc.), and several constants such as $\mathrm{i} = \sqrt{-1}$ and e can be obtained if we only have a single function $\mathrm{eml}(x, y) = \exp(x) - \ln(y)$ (and a constant 1). This function is known as exp-minus-log, or eml, for short.

For example, $e^x = \mathrm{eml}(x, 1)$, $\ln(x) = \mathrm{eml}(1, \mathrm{eml}(\mathrm{eml}(1, x), 1))$, $0 = \ln(1)$, $x - y = \mathrm{eml}(\ln(x), \exp(y))$, $x + y = x - (0 - y)$, $x/y = \exp(\ln(x) - \ln(y))$, $x \cdot y = x/(1/y)$, etc. etc.

Comment. The papers [7,8] list several other functions with the same universality property, e.g., $\mathrm{edl}(x, y) = \exp(x)/\ln(y)$, $\mathrm{lde}(x, y) = \ln(x)/\exp(y)$, $\mathrm{pli}(x, y) = (\ln(x))^{1/y}$, and $\mathrm{plm}(x, y) = (\ln(x))^{-y}$. Out of all these functions, eml is the fastest to compute – in the sense that its computation requires applying two basic functions of one variable – $\exp(x)$ and $\ln(y)$ – and the simplest (= fastest) possible arithmetic operation: addition/subtraction. All other listed universal functions require more computations. For example, since computing a^b means computing $\exp(b \cdot \ln(a))$, computing $(\ln(x))^{-y}$ means computing $\exp(-y \cdot \ln(\ln(x))$, which includes three applications of basic functions – two application of $\ln(x)$ and one application of $\exp(x)$ – plus multiplication.

This result can potentially speed up computations. Similar to how the result about NAND and NOR gates is used in computing Boolean expressions, this new result can be potentially useful in computing functions of several variables – and thus, in solving many practical problems.

Comments.

- Of course, the analogy with NAND gates is not perfect: by using NAND, we can implement any Boolean function, while by using eml, we can only implement a certain class of functions.
- It should also be mentioned that, as the author of [7,8] himself mentions, at the current moment, this speed up possibility remains largely a theoretical one, it is far from practical implementation. There are several serious problems that need to be solved before this can become practical. For example:
  - to get trigonometric functions, these papers use complex numbers – and for complex numbers, the value $\ln(x)$ is not determined uniquely: it is only determined modulo adding $2\pi\mathrm{i}$;

– while the eml approach if promising, it is not even 100% clear whether the use of eml functions will indeed speed up computations and if yes, in what cases; what is clear is that not in all cases, the use of eml representation leads to a speed-up; e.g., while subtraction can be represented via eml, direct subtraction requires much fewer bit operations than computing $\exp(x)$ and $\ln(y)$ needed to implement eml.

A natural question. From the practical viewpoint, an important question is which universal function should be implemented. Should it be elm? Should it be some other universal function – either one of the already discovered ones or maybe a not-yet-discovered one?

The answer to this question depends on what we want – i.e., what optimality criteria we use for this selection. For example, among the universal functions listed in [7,8], the eml function is the fastest to compute. However, the fact that elm is faster to compute than other tested universal functions does not mean that there are no other not-yet-discovered universal function which is even faster to compute.

It should be mentioned that faster computation time on the current computer architecture does not necessarily mean that in some future computer architecture, another universal function will not be faster. Also, depending on the application needs, other criteria may be important – e.g., ability to parallelize, space needed for computations, robustness to input uncertainty, etc.

It is therefore desirable to provide an answer to the above question – which universal function we should use – not only for computation time, but for all reasonable optimality criteria.

What we do in this paper. In this paper, we provide mathematical arguments that will hopefully help answer the above natural question. Specifically, we provide invariance-based arguments – and the corresponding preliminary results – about eml, arguments and results that will hopefully help prove that the eml function is indeed optimal (in some reasonable sense).

The structure of this paper. We start the paper by recalling, in Section 2, why we need data processing in the first place. Based on this analysis, in Section 3, we explain why symmetries naturally appear in such situations. In Section 4, we consider the two most natural transformations with respect to which data processing techniques should be invariant. We also emphasize that, depending on the practical situations, we need to distinguish between two types of invariances: invariance proper and covariance; this is done in Section 5. In Section 6, we explain the general relation between invariance and optimality. This relation leads to a description of optimal data processing functions of one variable in Section 7. In Section 8, we use invariances to find possibly optimal binary operations for combining functions of one variable – which leads us, in Section 9, to a class of functions containing the eml function. A brief conclusion forms Section 10.

2. Why we need data processing in the first place

General idea. The main purpose of elementary functions and their combinations is to process data, i.e., to process values of different physical quantities. So, to describe which functions are more adequate, it is important to recall why we need data processing in the first place.

What do we humans want? We want to know the current state of the world, we want to know the future state of the world, and we want to know what we can do to make the future state as beneficial for us as possible. The state of the world is characterized by the values of physical quantities. Some of these values we can measure directly: e.g., we can simply measure the distance between the two city locations. However, for many quantities

y, it is difficult to measure them directly. For example, it is difficult to directly measure the distance from the Earth to a faraway star. Since we cannot *directly* measure the value y, we need to measure it *indirectly*:

- we find some easier-to-measure quantities $x_1, \ldots, x_n$ that are related to y by a known dependence $y = f(x_1, \ldots, x_n)$;
- we measure these quantities and use the measurement results $\widetilde{x}_1, \ldots, \widetilde{x}_n$ to provide the following estimate for y:

$$\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n). \quad (1)$$

Computing this estimate is one of the cases of data processing.

Future values we cannot measure directly at all – we want to know these values now, before they can be measured. For these values, the only way to estimate a future value y is:

- to find some current (and past) values $x_1, \ldots, x_n$ that are related to y by a known dependence $y = f(x_1, \ldots, x_n)$,
- to measure these quantities, and then
- to use the measurement results $\widetilde{x}_i$ to estimate the desired future value y.

Similarly, to determine the parameters y_k of the best control or of the design strategy, we need to know how each of these parameters depends on the values x_i that describe the current state of the world – and then use data processing $y_k = f_k(x_1, \ldots, x_n)$ to estimate y.

Examples. Let us give two examples. The first example is related to the fact that we cannot (yet) directly measure the distance r to a galaxy – even to a neighboring one. To determine this distance, astronomers rely on the fact that some stars – known as *Cepheids* – have a periodically changing brightness, with some period T. Fr these stars, there is a known relation between the star's maximum actual brightness B and the period T: $B = f(T)$, for a known function $f(T)$. The observed brightness b of a star is inverse proportional to the square of its distance r from us: $b = c \cdot B(T)/r^2$, for a known constant c. Thus, we can compute r as $r = \sqrt{c \cdot B(T)/b}$. In this case, $y = r$, $n = 2$, $x_1 = T$, $x_2 = b$, and $f(x_1, x_2) = \sqrt{c \cdot B(x_1)/x_2}$.

Geosciences provide a more complex example of data processing. To find out what is happening at depth – whether there are mineral deposits or whether there is a dangerous fault that can cause earthquakes – researchers perform small explosions that cause seismic waves. These waves propagate through different layers of the Earth and can be detected (and measured) on the Earth surface, at different distances from the original explosion. Based on the results $x_{i,k}$ of measuring the amplitude of these waves at different locations i at different moments of time k, researchers come up with estimates for the desired quantities y – e.g., matter density at different depths.

3. Why symmetries naturally appear

How do we get all the knowledge? For example, how did our ancestors know that the Sun will rise in the morning? Because they observed it again and again. The seasons changed, thei behavior changed, their location may have changed, but the Sun continued to rise. So, they concluded that whether the Sun will rise does not depend on the location on the Earth, does not depend on the season, does not depend on their behavior. In mathematical terms, they concluded that the Sun-rising event is invariant with respect to changing the location, with respect to moving ahead in time, etc. This enabled them to predict that the Sun will continue rising in the future, no matter where they are.

How do we know that if we drop a pen, it will start falling down with the acceleration of 9.81 m/sec^2? Now we know it because it follows from the laws of gravity, but before Newton, how did people know that this will happen? Because people tried this in different locations on Earth, at different orientations, at different moments of time – and always

got the same result. So, they concluded that this phenomenon is invariant with respect to shifts in space and in time, and this enabled them to predict that this will always happen anywhere on Earth.

Starting with Newton, we know that the pen will start falling because of the laws of gravity. But how do we know the the laws of gravity are correct? Because these laws are true on Earth – with the apple falling on someone's head, they are true in space – because the motion of the Moon and planets satisfy the same laws, etc. In other words, these laws are invariant with respect to shifts and rotations in space (and, by the way, also with respect to scaling).

How do we know that Ohm's law $V = I \cdot R$ will be valid on Mars? Because people had many experiments at different locations, at different moments of time – and it always worked. So we can conclude that this law is valid everywhere.

This is the background of all our knowledge: we have some phenomenon that repeats in somewhat different situations, so we conclude that this phenomenon is invariant with respect to the corresponding changes – and thus, predict that it will remain the same under new similar changes – e.g., after time shift, i.e., in the future moments of time. So, invariances – that physicists call *symmetries* – indeed naturally appear as a foundation of all our knowledge.

Not only that, many laws of physics can be derived from the corresponding invariances; see, e.g., [2,9].

4. What are natural transformations

A large part of data processed by computers are values of different physical quantities. From this viewpoint, it is important to remember that the numerical value of a quantity depends on the choice of a measuring unit – and, for some quantities, on the selection of the starting value. For different measuring units and for different starting values, the same quantity is described by different numbers.

For example, the 2 meters length has a numerical value $2 \cdot 100 = 200$ when described in centimeters. The temperature of 35°C has a numerical value of $35 + 273.15 = 308.15$ when described in Kelvins – a scale that has a different starting point.

In general, if we replace the original measuring unit by a different unit which is λ times smaller, then each numerical value x is multiplied by λ: $x \to x' = \lambda \cdot x$. In particular, for a transition from meters to centimeters, $\lambda = 100$.

Similarly, if we replace the original starting point with a new starting point which is c units before the original one, then c is added to all the numerical values: $x \to x' = x + c$. In particular, for a transition from Celsius to Kelvin temperature scales, $c = 273.15$.

5. Invariance vs. covariance

In the previous text, when we talked about the fact that some physical processes do not change under appropriate transformations, we called such situations *invariances*. However, strictly speaking, we need to distinguish between two types of such situations:

- There are invariances in the narrow sense of this word, meaning that the values of some physical quantities do not change when we apply the corresponding transformation. For example, the acceleration of the free falling object does not change if we move this object to a nearby location on Earth.
- There are also what are called *covariances*, when the physical process remains the same, but the values of the corresponding quantities need to be adjusted. For example, when we test a scaled model of an airplane in a wind tunnel – to analyze how a proposed airplane design will work – the airflows of the model and of the airplane are (largely)

similar, but the aerodynamic forces acting on the model need to be appropriately re-scaled to get the forces acting on the actual airplane.

Let us give a precise example. Suppose that we are analyzing how the temperature x changes with time t, and a transformation is shift by one year: $t \mapsto t+1$. Invariance would means that for each moment of time, we have $x(t+1) = x(t)$, i.e., the same temperature patterns repeats year after year. Covariance – with respect to an appropriate shift of temperatures $x \mapsto x + x_0$ – means that $x(t+1) = x(t) + x_0$, i.e., that, in additional to seasonal fluctuations, there is also a warming trend, based on which the temperature increases by x_0 degrees every year.

As of now, this difference may sound purely terminological, but it is actually useful: we will use it in our explanation of the eml.

6. How are invariances related to optimality: a simple general result

What does "optimal" mean: discussion. In many cases, "optimal" means that we have some objective function $f(x)$ that we want to maximize (or minimize), and an alternative a_{opt} is optimal if $f(a_{\text{opt}}) \geq f(a)$ for all other alternatives a (or, in case of minimization, when $f(a_{\text{opt}}) \leq f(a)$ for all other alternatives a).

However, this is not the only way to describe optimality. Sometimes, for a given objective function $f(x)$, there are several different alternatives that are all optimal. For example, if we are looking for the best sorting algorithm, and our objective function is the average computation time, then several sorting algorithms are optimal: quicksort, mergesort, heapsort, etc.; see, e.g., [1]. If we are looking for the best faculty candidate for a position, we may have several candidates whose research records are equally good.

In such situations, it is reasonable to use this non-uniqueness to optimize something else. For example, we may want to select, among all pre-selected sorting algorithms, one with the smallest worst-case computation time. Among all pre-selected faculty candidates, we may want to select one who is the best teacher, etc. In this case, the optimality criterion is no longer described by a single objective function. In this case, an alternative a is better than the alternative b is either $f(a) > f(b)$ or ($f(a) = f(b)$ and $g(a) > g(b)$) for some other function $g(x)$.

If after this clarification, we still have several equally good alternatives, this means that our optimality criterion is not final: we can use the remaining non-uniqueness to optimize something else. When only one optimal alternative is left, this means that we have reached the final optimality criterion.

On all the stages, what is important is to have some way to compare two alternatives a and b and decide whether a is better, or b is better, or that these alternatives are of equal quality. Let us denote "a is better than b" by $a \succ b$, and "a and b are of equal quality" by $a \sim b$. These two relations have to be consistent: e.g., if a is better than b and b is better than c, then a should be better than c.

What does "optimal" mean: definitions. Thus, we arrive at the following definitions.

Definition 1. *Let A be a set. Its elements will be called* alternatives. *By an* optimality criterion, *we mean a pair of binary relations $(\succ, \sim)$ that satisfy the following properties for all a, b, and c:*

- *if $a \succ b$ and $b \succ c$, then $a \succ c$;*
- *if $a \succ b$ and $b \sim c$, then $a \succ c$;*
- *if $a \sim b$ and $b \succ c$, then $a \succ c$;*
- *if $a \sim b$ and $b \sim c$, then $a \sim c$;*
- *if $a \sim b$, then $b \sim a$;*
- *if $a \succ b$, then we cannot have $a \sim b$;*
- *$a \sim a$.*

Definition 2. *We say that an alternative a_{opt} is* optimal *if for every $a \in A$, we have either $a_{\text{opt}} \succ a$ or $a_{\text{opt}} \sim a$.*

Definition 3. *We say that the optimality criterion is* final *if there is exactly one optimal alternative.*

Invariances. In many practical situations, there are some transformations that do not change which alternative is better. For example, when we are looking for investment alternatives and we want to select the most profitable one, it does not matter whether we measure the profit in US dollars, in Euros, on in yen. Changing from one monetary unit to another one means multiplying all numerical values by an appropriate constant λ: $x \mapsto \lambda \cdot x$. In this case, these transformations are invariances. As we have mentioned, in physics, such invariances are called *symmetries*; see, e.g., [2,9].

If the comparison does not change when we apply a transformation g_1, and does not change when we apply another transformation g_2, then, of course, it will not change if we first apply g_1 and then apply g_2, i.e., equivalently, if we apply the composition $g_2(g_1)$. Similarly, the comparison will not change if we apply the inverse transformation g^{-1}. Thus, the class of such transformations must be closed under composition and under taking an inverse. In mathematics, such classes are known as *transformation groups*. So, we arrive at the following definition.

Definition 4. *Let G be a transformation group on the set A. We say that an optimality criterion $(\succ, \sim)$ is* G-invariant *for all $a, b \in A$ and for all $g \in G$, the following two properties are satisfied:*

- *if $a \succ b$ then $g(a) \succ g(b)$, and*
- *if $a \sim b$ then $g(a) \sim g(b)$.*

Comment. In different situations, we have different optimality criteria. The intended criteria may be computational simplicity, implementation cost, expression depth, numerical stability, runtime, etc. In many practical situations, these criteria are invariant with respect to some reasonable transformations. For example, if we consider average runtime over many practical problems, it should not natter whether we describe lengths in meters or in inches or in centimeters.

Some of these criteria may be final – in the sense that for the corresponding criterion, there is only one optimal alternative. Some of these criteria may turn out to be not final – in the sense that there are several different optimal alternatives. In this case, there is an opportunity to use some additional criterion – in other words, to replace the original non-final criterion with a final one.

In this paper, instead of selecting a *single* optimality criterion and analyzing which alternatives are optimal with respect to this criterion, we consider a more general approach. Namely, we analyze which alternatives may be optimal under *all possible* invariant final optimality criteria.

How is invariance related to optimality. This relation is described by the following simple result:

Proposition 1. [6] *For every G-invariant final optimality criterion, the optimal alternative a_{opt} is also G-invariant, i.e., $g(a_{\text{opt}}) = a_{\text{opt}}$ for all $g \in G$.*

Proof. By definition of optimality, for every $a \in A$, we have $a_{\text{opt}} \succ a$ or $a_{\text{opt}} \sim a$. In particular, for every $g \in G$, we have $a_{\text{opt}} \succ g^{-1}(a)$ or $a_{\text{opt}} \sim g^{-1}(a)$. Since the optimality criterion is G-invariant, this implies that $g(a_{\text{opt}}) \succ a$ or $g(a_{\text{opt}}) \sim a$. By definition of an optimal alternative, this means that $g(a_{\text{opt}})$ is optimal. But since the optimality criterion is final, there is only one optimal alternative. So indeed, $g(a_{\text{opt}}) = a_{\text{opt}}$. The proposition is proven.

Conclusion of this section. For final invariant criteria, to find an optimal alternative, it is sufficient to search among invariant alternatives. This narrows down the search and thus, hopefully, makes finding the optimal solution easier.

Comments.

- This idea has been used in many computer-related applications; see, e.g., [6].
- Of course – unless there is only one invariant alternative – Proposition 1 does not solve the optimization problem. All we can say is that the optimal alternative is among the invariant ones, but we do not automatically know which of them is optimal.
- The physical meaning of Proposition 1 is straightforward. If we have an object that does not change when we rotate it around some axis, and we know that this object has only one highest point, then this point is located on the rotation axis. Indeed, if this point was not on this axis, then all the points obtained from this one by rotation would also have the same (largest) height – but we assumed that there is only one highest point.
- In the following sections, we will use this result to narrow down the search for optimal data processing operations.

7. Optimal functions of one variable

What we do in this section. In this section, we use the above-described invariance ideas to come up with the list of possibly optimal functions $y = f(x)$ of one variable. In this section, we consider non-constant functions from a non-empty non-single-element set of real numbers to real numbers.

Analysis of the problem. In one of the previous sections, we mentioned two natural transformations; scaling (that corresponds to changing the measuring unit) and shift (that corresponds to changing the starting point). If we consider the possibility to change the measuring unit, then the same dependence $y = f(x)$ – expressed in some y-units – will take the form $C \cdot f(x)$ if we use a different measuring unit – which is C times smaller. Since these two functions describe the exact same physical dependence, it makes sense to consider not individual functions $f(x)$, but whole families $\{C \cdot f(x)\}_{C>0}$ in which the function $f(x)$ is fixed but the coefficient C can take any positive value.

Similarly, if we consider the possibility to change the starting point for measuring y, then the same dependence $y = f(x)$ – expressed by using some starting point for y-measurements – will take the form $f(x) + C$ if we use a different starting point – a starting point which is C units earlier. Since these two functions describe the exact same physical dependence, it makes sense to consider not individual functions $f(x)$, but whole families $\{f(x) + C\}_C$ in which the function $f(x)$ is fixed but the coefficient C can take any real value.

For simplicity, let us assume that in both cases, the dependence $f(x)$ is smooth, i.e., differentiable. We can similarly consider scaling and shift of the quantity x. Thus, we arrive at the following formulations – that we will list together with results.

Definition 5.

- *By a* scaled family, *we mean the family* $\{C \cdot f(x)\}_{C>0}$, *where* $f(x)$ *is a differentiable non-constant real-valued function defined on a non-empty non-single-element set of real numbers, and the coefficient* C *can take any positive value.*
- *By a* shifted family, *we mean the family* $\{f(x) + C\}_C$, *where* $f(x)$ *is a differentiable non-constant real-valued function defined on a non-empty non-single-element set of real numbers, and the coefficient* C *can take any real value.*
- *Let* $\lambda > 0$.

- *By the* λ-scaling *of a function* $F(x)$*, we mean the function* $S_\lambda(F)$ *defined as* $(S_\lambda(F))(x) = F(\lambda \cdot x)$.
- *By the* λ-scaling *of a family of functions* $\mathcal{F}$*, we mean the family*

$$S_\lambda(\mathcal{F}) \stackrel{\text{def}}{=} \{S_\lambda(F) : F \in \mathcal{F}\}.$$

- *We say that an optimality criterion on the set of families of functions is* scale-invariant *if it is invariant with respect to all* λ*-scalings.*

- *Let* c *be a real number.*
  - *By the* c-shift *of a function* $F(x)$*, we mean the function* $T_c(F)$ *defined as* $(T_c(F))(x) = F(x + c)$.
  - *By the* c-shift *of a family of functions* $\mathcal{F}$*, we mean the family*

$$T_c(\mathcal{F}) \stackrel{\text{def}}{=} \{T_c(F) : F \in \mathcal{F}\}.$$

  - *We say that an optimality criterion on the set of families of functions is* shift-invariant *if it is invariant with respect to all* c*-shifts.*

Proposition 2. *Let* $\mathbb{R}$ *denote the set of all real numbers, and for each property* P*, let*

$$\mathbb{R}_P \stackrel{\text{def}}{=} \{x \in \mathbb{R} : P(x)\}.$$

For example, $\mathbb{R}_{>0}$ *is the set of all positive numbers.*

- *For every scale-invariant final optimality criterion on the set of all scaled families, each function* $f(x)$ *from the optimal family has the following form, for some* α*. This function* $f(x)$ *is defined on one of the sets* $\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{<0}$, $\mathbb{R}_{\leq 0}$, $\mathbb{R}_{\neq 0}$*, and on each of the sets* $\mathbb{R}_{>0}$ *and* $\mathbb{R}_{<0}$ *on which it is defined, it has the form* $f(x) = C \cdot x^\alpha$.
- *For every shift-invariant final optimality criterion on the set of all scaled families, the optimal class corresponds to* $f(x) = \exp(a \cdot x)$ *for some* a.
- *For every scale-invariant final optimality criterion on the set of all shifted families, each function* $f(x)$ *from the optimal family has the following form, for some* $a \neq 0$*. This function* $f(x)$ *is defined on one of the sets* $\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{<0}$, $\mathbb{R}_{\leq 0}$, $\mathbb{R}_{\neq 0}$*, and on each of the sets* $\mathbb{R}_{>0}$ *and* $\mathbb{R}_{<0}$ *on which it is defined, it has the form* $f(x) = a \cdot \ln(|x|) + C$.
- *For every shift-invariant final optimality criterion on the set of all shifted families, the optimal class corresponds to* $f(x) = a \cdot x$ *for some* a.

Physical examples. Before proceeding to the proofs, let us provide physical examples of the first two types of dependence.

- The first case can be illustrated on the example of the energy spectrum $f(\omega)$ in faraway radio astronomy. Most faraway astronomical radio waves come from high-energy electrons, electrons with energy much larger than their rest mass. At such energies, the rest mass can be safely ignored. So, we can safely assume that there is no fixed energy. Thus, physics remains the same if we use a different unit to measure energy, i.e., if we replace the numerical values of energy E to $\lambda \cdot E$. Thus, the numerical values of frequency ω (that are related to E due to the quantum formula $E = \hbar \cdot \omega$) change to to $\lambda \cdot \omega$. Of course, if we change the units of energy and frequency, we must accordingly change the unit of energy density, to a one which is $c(\lambda)$ smaller. The resulting invariance $f(\lambda \cdot \omega) = c(\lambda) \cdot f(\omega)$, as we show in our proof of Proposition 2, leads to the power law $f(\omega) = C \cdot \omega^\alpha$ – which is exactly what is observed.
- The second case can be illustrated by the dependence of atmospheric pressure $p(h)$ on the height h above Earth. In general, there is no fixed starting point for height –

some points on the Earth surface are lower, some are higher. So, it makes sense to assume that the correspond physics should not change if we change the starting point, i.e., go from h to $h + h_0$. Of course, the pressure decreases with hieght, so to maintain the same dependence, we should also re-calibrate pressure, from p to $c(h_0) \cdot p$ for an appropriate scaling parameter $c(h)$. The resulting invariance $p(h + h_0) = c(h_0) \cdot p(h)$ leads – as the following proof of Proposition 2 shows – to the observed exponential-law dependence $p(h) = C \cdot \exp(-k \cdot h)$ for some C and k.

Of course, in both cases, the symmetry-based dependencies are only approximate. For example, the power law breaks down when we consider frequencies for which the energy is comparable with the electron's rest mass – and thus, this rest mass can no longer be ignored.

Comment. In the last three cases, the simplest classes correspond to $a = 1$, i.e., to the functions $\exp(x)$, $\ln(x)$, and x.

Proof of Proposition 2. For all four results, Proposition 1 implies that the optimal family is itself invariant.

1°. For the first case, invariance means that $\{C \cdot f(\lambda \cdot x)\}_{C>0} = \{C \cdot f(x)\}_{C>0}$. In particular, this implies that the function $f(\lambda \cdot x)$ from the first family belongs to the second family, i.e., that

$$f(\lambda \cdot x) = C(\lambda) \cdot f(x) \tag{2}$$

for some $C(\lambda)$.

By definition of a scaled family, the function $f(x)$ is defined for at least two different values x. This means that at least one of these values x_0 if different from 0, so this value either positive or negative. The equality (3) implies that it is defined for the values $x_0 + c$ corresponding to all possible real numbers c.

So, if the function $f(x)$ is defined for some positive value $x_0 > 0$, then for each positive real number x, we can take $\lambda = x/x_0$, and conclude that the function $f(x)$ is defined for this x. Thus, if the function $f(x)$ is defined for *some* $x_0 > 0$, it is defined for *all* positive values x. Similarly, when $f(x)$ is defined for at least *one* negative x, then it is defined for *all* negative x.

Also, if for some positive x_0, we have $f(x_0) \neq 0$, then $f(x) \neq 0$ for all $x > 0$. Indeed, if we had $f(x_1) = 0$ for some $x_1 > 0$, then from (3), with $x = x_1$ and $\lambda = x_0/x_1$, we would conclude that $f(x_0) = C(x_0/x_1) \cdot f(x_1) = 0$, while we know that $f(x_0) \neq 0$. Thus, all the values of the function $f(x)$ for $x > 0$ are different from 0. Similarly, if the function $f(x)$ is different from 0 for *at least one* negative value x, then its values are non-zeros for *all* negative x.

Let us first consider the case when $f(x)$ is different from 0 for all positive values x. Both expressions $f(\lambda \cdot x)$ and $f(x)$ are differentiable, and $f(x) \neq 0$, so their ratio $C(\lambda)$ is also differentiable. So, we can differentiate both side of the formula (2) with respect to λ and get $x \cdot f'(\lambda \cdot x) = C'(\lambda) \cdot f(x)$. In particular, for $\lambda = 1$, we get $x \cdot f'(x) = \alpha \cdot f(x)$, where we denoted $\alpha \stackrel{\text{def}}{=} C'(1)$, i.e.:

$$x \cdot \frac{df}{dx} = \alpha \cdot f.$$

We can separate the variables if we divide both sides by x and by f, and multiply both sides by dx. Then, we get

$$\frac{df}{f} = \alpha \cdot \frac{dx}{x}.$$

Integrating both sides, we get $\ln(f) = \alpha \cdot \ln(x) + C_1$, where C_1 is the integration constant. Applying $\exp(x)$ to both sides, we get:

$$f(x) = e^{\alpha \cdot \ln(x) + C_1} = e^{\alpha \cdot \ln(x)} \cdot e^{C_1} = \left(e^{\ln(x)}\right)^{\alpha} \cdot e^{C_1} = C \cdot x^{\alpha},$$

where we denoted $C \stackrel{\text{def}}{=} e^{C_1}$. For this function, the formula (2) is satisfied for $C(\lambda) = \lambda^{\alpha}$.

Similarly, if the function $f(x)$ is non-zero for negative x, we conclude that $f(x) = C \cdot x^{\alpha_-}$ for some α_-, with $C(\lambda) = \lambda^{\alpha_-}$. (In this case, instead of $\ln(x)$, integration will lead to $\ln(|x|)$.) If the function $f(x)$ is defined and non-zero for both positive and negative x, then from the fact that the same $C(\lambda)$ should be used for positive and negative x, we conclude that $\alpha_- = \alpha$.

This part of the proposition is thus proven.

2°. For the second case, invariance means that $\{C \cdot f(x+c)\}_{C>0} = \{C \cdot f(x)\}_{C>0}$. In particular, this implies that the function $f(x+c)$ from the first family belongs to the second family, i.e., that

$$f(x+c) = C(c) \cdot f(x) \tag{3}$$

for some $C(c)$.

By definition of a scaled family, the function $f(x)$ is defined at some point x_0. The equality (3) implies that $f(x)$ is defined for the values $x_0 + c$ corresponding to all possible real numbers c. In particular, for each real number x, we can take $c = x - x_0$, and conclude that the function $f(x)$ is defined for this x. Thus, the function $f(x)$ is everywhere defined.

Also, by definition of a scaled family, the function $f(x)$ is not a constant. In particular, this means that not all its values are 0s, so there must exist a value x_0 for which $f(x_0) \neq 0$. Let us show that it implies that $f(x) \neq 0$ for all x. Indeed, if we had $f(x_1) = 0$ for some x_1, then from (3), with $x = x_1$ and $c = x_0 - x_1$, we would conclude that

$$f(x_0) = C(x_0 - x_1) \cdot f(x_1) = 0,$$

while we know that $f(x_0) \neq 0$. Thus, all the values of the function $f(x)$ are different from 0.

Both expressions $f(x+c)$ and $f(x)$ are differentiable, so their ratio $C(c)$ is also differentiable. So, we can differentiate both side of the formula (3) with respect to c and get $f'(x+c) = C'(c) \cdot f(x)$. In particular, for $c = 0$, we get $f'(x) = a \cdot f(x)$, where we denoted $a \stackrel{\text{def}}{=} C'(0)$, i.e.,

$$\frac{df}{dx} = a \cdot f.$$

We can separate the variables if we divide both sides by f and multiply both sides by dx. Then, we get

$$\frac{df}{f} = a \cdot dx.$$

Integrating both sides, we get $\ln(f) = a \cdot x + C_1$, where C_1 is the integration constant. Applying $\exp(x)$ to both sides, we get

$$f(x) = \exp(a \cdot x + C_1) = C \cdot \exp(a \cdot x),$$

where we denoted $C \stackrel{\text{def}}{=} \exp(C_1)$. The second part of the proposition is also proven.

3°. For the third case, invariance means that $\{f(\lambda \cdot x) + C\}_C = \{f(x) + C\}_C$. In particular, this implies that the function $f(\lambda \cdot x)$ from the first family belongs to the second family, i.e., that

$$f(\lambda \cdot x) = f(x) + C(\lambda) \tag{4}$$

for some $C(\lambda)$.

Similarly to Part 1 of this proof, we can prove that:

- if the function $f(x)$ is defined for *some* positive value x, then it is defined for *all* positive x, and
- if $f(x)$ is defined for *some* negative x, then it is defined for *all* negative x.

Let us consider the case when $f(x)$ is defined for all positive x. Both the expressions $f(\lambda \cdot x)$ and $f(x)$ are differentiable, so their difference $C(\lambda)$ is also differentiable. So, we can differentiate both side of the formula (4) with respect to λ and get $x \cdot f'(\lambda \cdot x) = C'(\lambda)$. In particular, for $\lambda = 1$, we get $x \cdot f'(x) = a$, where we denoted $a \stackrel{\text{def}}{=} C'(1)$, i.e.,

$$x \cdot \frac{df}{dx} = a.$$

We can separate the variables if we divide both sides by x and multiply both sides by dx. Then, we get

$$df = a \cdot \frac{dx}{x}.$$

Integrating both sides, we get $f(x) = a \cdot \ln(x) + C_1$, where C_1 is the integration constant. For this function, the equality (4) holds for $C(\lambda) = a \cdot \ln(\lambda)$.

Similarly, if the function $f(x)$ is defined for all negative x, then for these x, we have $f(x) = a_- \cdot \ln(|x|) + c_1$ for soem a_- and C_1. If the function $f(x)$ is defined for both positive and negative x, then from the fact that the same $C(\lambda)$ should be used for positive and negative x, we conclude that $a_- = a$.

So, this part of the proposition is also proven.

4°. For the fourth case, invariance means that $\{f(x+c) + C\}_C = \{f(x) + C\}_C$. In particular, this implies that the function $f(x+c)$ from the first family belongs to the second family, i.e., that

$$f(x+c) = f(x) + C(c) \tag{5}$$

for some $C(c)$.

Similarly to Part 2 of this proof, we can thus prove that the function $f(x)$ is everywhere defined. Both the expressions $f(x+c)$ and $f(x)$ are differentiable, so their difference $C(c)$ is also differentiable. So, we can differentiate both side of the formula (5) with respect to c and get $f'(x+c) = C'(c)$. In particular, for $c = 0$, we get $f'(x) = a$, where we denoted $a \stackrel{\text{def}}{=} C'(0)$.

Integrating both sides, we get $f(x) = a \cdot x + C_1$, where C_1 is the integration constant. The function $f(x)$ cannot be constant, so $a \neq 0$. The last part of the proposition is thus proven, and so is the proposition.

8. Optimal binary operations

What we do in this section: a reminder. We would like to have a function of two variables which is, in some reasonable sense, the best candidate for the desired universal property – the property that many basic functions can be represented by superposition of such functions. This property does not seem to depend on the choice of a measuring unit or on a choice of the starting point.

To construct such a function, it is reasonable to apply an optimal binary operation to optimal functions of one variable – the ones that we described in the previous section. As planned, to find the optimal binary operations, we will use the above result relating optimality and invariance.

Let us find invariant binary operations. To describe what we mean by invariant binary operations, let us recall the difference between invariance proper and covariance.

Definition 6. *By a* binary operation *(or simply* operation*, for short), we mean a differentiable function of two variables $f(y_1, y_2)$ defined for all pairs of real numbers.*

- *We say that an operation $f(y_1, y_2)$ is* shift-invariant *if for all C and for all y_1 and y_2, we have $f(y_1 + C, y_2 + C) = f(y_1, y_2)$.*
- *We say that an operation $f(y_1, y_2)$ is* shift-covariant *if for all C and for all y_1 and y_2, we have $f(y_1 + C, y_2 + C) = f(y_1, y_2) + C$.*
- *We say that an operation $f(y_1, y_2)$ is* scale-invariant *if for all $C > 0$ and for all y_1 and y_2, we have $f(C \cdot y_1, C \cdot y_2) = f(y_1, y_2)$.*
- *We say that an operation $f(y_1, y_2)$ is* scale-covariant *if for all $C > 0$ and for all y_1 and y_2, we have $f(C \cdot y_1, C \cdot y_2) = C \cdot f(y_1, y_2)$.*

Discussion. We want our binary operation $f(y_1, y_2)$ to be both shift- and scale-invariant. According to Definition 6, each of these two invariances can be interpreted in two different ways. So, we get $2 \cdot 2 = 4$ possible interpretations of the need to have both invariances. The following proposition describes what happens if we consider all interpretations.

Proposition 3.

- *The only shift-invariant and scale-invariant binary operations are constants $f(y_1, y_2) =$* const.
- *The only shift-invariant and scale-covariant binary operations are*

$$f(y_1, y_2) = C \cdot (y_1 - y_2)$$

  for some constant C.
- *No binary operation is shift-covariant and scale-invariant.*
- *The only shift-covariant and scale-covariant binary operations are*

$$f(y_1, y_2) = C \cdot y_1 + (1 - C) \cdot y_2,$$

  for some constant C.

Comments.

- Due to Proposition 1, for each final invariant optimality criterion on the set of all binary operations, the optimal binary operations should take one of the above forms.
- It is worth mentioning that for the values C between 0 and 1, the last binary operation – known as *Hurwicz combination* after the Nobel prize winner who invented it – is exactly what decision theory recommends to use as a utility of an alternative if all we know is that the actual utility is located somewhere between y_1 and y_2; see, e.g., [3–5].
- According to Proposition 3, the only non-trivial operations are difference and Hurwicz combination. From the computational viewpoint, the difference $y_1 - y_2$ is simpler (it requires only one hardware supported operation), so our preference is on the difference – but maybe Hurwicz combination, when applied to $\exp(x)$ and $\ln(y)$, will also lead to a universal function. We will discuss this in Section 9.

Proof of Proposition 3.

1°. Let us first consider the case of shift-invariance and scale-invariance. It is easy to check that constants are both shift-invariant and scale-invariant. So, to prove this part of the proposition, we need to prove that every shift-invariant and scale-invariance operation is a constant.

Indeed, shift-invariance means that we always have $f(y_1 + C, y_2 + C) = f(y_1, y_2)$. In particular, for $C = -y_2$, we conclude that $f(y_1, y_2) = f(y_1 - y_2, 0)$, i.e., that $f(y_1, y_2) = F(y_1 - y_2)$, where we denoted $F(y) \stackrel{\text{def}}{=} f(y, 0)$. Now, for $y_2 = 0$, scale-invariance means that $f(C \cdot y_1, 0) = f(y_1, 0)$, i.e., that $F(C \cdot y_1) = F(y_1)$. Now:

- for $y_1 > 0$, we can take $C = 1/y_1$ and thus conclude that $F(y_1) = F(1)$;
- for $y_1 < 0$, we can take $C = -1/y_1$ and thus conclude that $F(y_1) = F(-1)$.

The function $F(y_1)$ is differentiable hence continuous. So:

- when we take $y_1 > 0$ and $y_1 \to 0$, we conclude that, in the limit, that $F(0) = F(1)$;
- when we take $y_1 < 0$ and $y_1 \to 0$, we conclude that, in the limit, that $F(0) = F(-1)$.

Since both $F(1)$ and $F(-1)$ are equal to the same number $F(0)$, they are thus equal, so $F(y_1) = F(1)$ for all y_1. So, the value $f(y_1, y_2) = F(y_1 - y_2)$ is also always equal to the constant $F(1)$. The first part of Proposition 3 is proven.

2°. Let us now consider the case of shift-invariance and scale-covariance. It is easy to check that each operation $C \cdot (y_1 - y_2)$ is both shift-invariant and scale-covariant. So, to prove this part of the proposition, we need to prove that every shift-invariant and scale-covariant operation has this form.

Similarly to Part 1 of this proof, from shift-invariance, we conclude that $f(y_1, y_2) = F(y_1 - y_2)$, where $F(y) \stackrel{\text{def}}{=} f(y, 0)$. For $y_2 = 0$, scale-covariance means that $f(C \cdot y_1, 0) = C \cdot f(y_1, 0)$, i.e., that $F(C \cdot y_1) = C \cdot F(y_1)$. Now:

- for $y_1 > 0$, we can take $C = 1/y_1$ and thus conclude that $F(1) = (1/y_1) \cdot F(y_1)$, hence $F(y_1) = F(1) \cdot y_1$;
- for $y_1 < 0$, we can take $C = -1/y_1$ and thus conclude that $F(-1) = (-1/y_1) \cdot F(y_1)$, hence $F(y_1) = -F(-1) \cdot y_1$.

The function $F(y_1)$ is differentiable hence continuous, so in the limit, we get $F(0) = 0$.

In particular, the function $F(y_1)$ is differentiable for $y_1 = 0$, i.e., we have a limit

$$F'(0) = \lim_{y_1 \to 0} \frac{F(y_1) - F(0)}{y_1}.$$

- when we take $y_1 > 0$ and $y_1 \to 0$, we conclude that, in the limit, that $F'(0) = F(1)$;
- when we take $y_1 < 0$ and $y_1 \to 0$, we conclude that, in the limit, that $F'(0) = -F(-1)$.

Since both $F(1)$ and $-F(-1)$ are equal to the same number $F'(0)$, they are thus equal, so $F(y_1) = F(1) \cdot y_1$ for all y_1. Thus, the value $f(y_1, y_2) = F(y_1 - y_2) = F(1) \cdot (y - y_2)$ has the desired form for $C = F(1)$. The second part of Proposition 3 is proven.

3°. Let us prove, by contradiction, that no operation can be both shift-covariant and scale-invariant.

Indeed, let $f(y_1, y_2)$ be such an operation. For $C = -y_2$, shift-covariance implies that $f(y_1 - y_2, 0) = f(y_1, y_2) - y_2$, i.e., that $f(y_1, y_2) = f(y_1 - y_2, 0) + y_2$. In particular, $f(C \cdot y_1, C \cdot y_2) = f(C \cdot (y_1 - y_2), 0) + C \cdot y_2$. Thus, scale-invariance means that

$$f(C \cdot (y_1 - y_2), 0) + C \cdot y_2 = f(y_1 - y_2, 0) + y_2.$$

In particular, for $y_1 = y_2 + 1$, we get $y_1 - y_2 = 1$, so $f(C, 0) + C \cdot y_2 = f(1, 0) + y_2$, i.e., equivalently, $f(C, 0) - f(1, 0) = (1 - C) \cdot y_2$. For $C = 0.5$, this implies that $f(0.5, 0) - f(1, 0) = 0.5 \cdot y_2$, i.e., $y_2 = 2 \cdot (f(0.5, 0) - f(1, 0))$. This must be true for all y_2, but if we

take a value y_2 which is different from the right-hand side, this equality is no longer true. So, the assumption that such an operation is possible leads to a contradiction. The third part of the proposition is also proven.

4°. Finally, let us consider the case of shift-covaraince and scale-covarience. It is easy to check that all the operations $C \cdot y_1 + (1 - C) \cdot y_2$ are both shift-covariant and scale-covariant. So, to prove this part of the proposition, we need to prove that every shift-covariant and scale-covariant operation has this form.

Indeed, similarly to Part 3 of this proof, shift-covariance implies that

$$f(y_1, y_2) = f(y_1 - y_2, 0) + y_2.$$

For this expression, scale-covariance leads to $f(C \cdot (y_1 - y_2), 0) + C \cdot y_2 = C \cdot f(y_1 - y_2, 0) + C \cdot y_2$. Subtracting $C \cdot y_2$ from both sides, we conclude that $F(C \cdot (y_1 - y_2)) = C \cdot F(y_1 - y_2)$, where we denoted $F(y) \stackrel{\text{def}}{=} f(y, 0)$. From Part 2 of this proof, we know that this implies that $F(y) = C \cdot y$ for some constant C. Thus,

$$f(y_1 - y_2) = F(y_1 - y_2) + y_2 = C \cdot (y_1 - y_2) + y_2 = C \cdot y_1 + (1 - C) \cdot y_2.$$

This part is also proven, and so is the proposition.

9. Let us apply optimal binary operations to optimal functions of one variable

Up to now, we have been proving mathematical results. Now, we would like to provide informal arguments that these results may lead to an explanation of why among possible universal functions, eml is optimal in some reasonable sense.

Our argument is that a natural way to get an optimal function of two variables is to apply an optimal binary operation $f(y_1, y_2)$ to two optimal functions $f_1(x_1)$ and $f_2(x_2)$, resulting in $g(x_1, x_2) = f(f_1(x_1), f_2(x_2))$. Which operation should we apply and to which functions of one variable?

Out of the possibly optimal operations listed in the previous section, which one should we apply? A constant operation does not make any sense: we just get a constant function $g(x_1, x_2) = \text{const}$, based on which we cannot compute anything. So, we are left with subtraction and Hurwicz combination. Maybe Hurwicz combination will also work, but clearly, subtraction is, from the computational viewpoint, the simplest of the two.

To which of the optimal functions of one variables should we apply our binary operation? In line with the above result relating optimality with invariance, optimality implies that we want a function that is both scale- and shift-invariant. In this requirement both scale- and shift-invariances appear in a similar way, so it makes sense to select a pair of optimal functions of one variable in such a way that this pair will be invariant with respect to swapping scale- and shift-invariance. One can see that there are only two such pairs:

- if we pick one of the functions to be shift-invariant under shifts of the input, the second function should be scale-invariant with respect to scaling of the input; for this pair, we need to combine x and x^α;
- similarly, if we pick one of the functions to be scale-invariant under shift of the input, the second function should be shift-invariant with respect to scaling of the input; for this pair, we need to combine $\exp(x)$ and $\ln(x)$.

From the commonsense viewpoint, the second options looks more promising, since:

- in the first pair, one of the function is the identity $f_1(x_1) = x_1$ (or a simple linear function), while
- in the second pair, both functions are non-trivial.

Also, if we consider x and x^α, we cannot get basic functions such as $\exp(x)$, $\ln(x)$, etc.

We can apply each of the two optimal binary operations – difference and Hurwicz combination – to each of these two pairs. We thus get $2 \cdot 2 = 4$ possible functions of two variables. Out of these four, if we apply the simplest optimal binary operation to the most promising pair of optimal functions of one variable, we get exactly the desired function $\exp(x) - \ln(y)$. This way, we have indeed provided some explanation of this function's success.

10. Conclusions

In many practical situations, having a universal tool is more efficient that having many specialized tools. This is true in many application areas, in particular, this is true in computing. For example, the fact that all possible Boolean operations can be performed if we only have a single type of gates – NAND gates – made computer design much more efficient. In contrast to hardware, for software we still have a lot of independently supported operations: arithmetic operations, special functions like $\sin(x)$ and $\exp(x)$, and different special constants like e and π. A recent breakthrough paper has shown that, in principle, all these operations can be performed if we implement only one binary operation – the function $\exp(x) - \ln(y)$ known as eml. This result may potentially lead to faster computations. Faster computations may, in their turn, enable us to solve more complex problems – problems that now require too much computation time to be practically solvable.

The function $\mathrm{eml}(x, y) = \exp(x) - \ln(y)$ is not the only one with the above universal property, several others have been proposed. Of the ones that have been proposed, eml is the fastest to compute. However, without a theoretical analysis we cannot be sure that some other – yet untested – universal function will work even better. In this paper, we use the symmetry technique – common in physics and in many other applications – to provide arguments that, in some reasonable sense, eml is probably the optimal function. This results provides an additional encouragement for plans of designing promising eml-based computational devices.

Author Contributions: All the authors performed the research described in this article and editing of the text. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the AT&T Fellowship in Information Technology, by the Institute for Risk and Reliability, Leibniz Universitaet Hannover, Germany, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Focus Program SPP 100+ 2388, Grant Nr. 501624329, by the European Union under the project ROBOPROX (No. CZ.02.01.01/00/22 008/0004590), by the Center of Excellence in Econometrics, Faculty of Economics, Chiang Mai University, Thailand, by the Ho Chi Minh City University of Banking, Vietnam, and by Thang Long University, Hanoi, Vietnam.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors are greatly thanks to the anonymous referees for valuable suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cormen, Th.H., Leiserson, C.E., Rivest, R.L. Stein, C. *Introduction to Algorithms*, MIT Press: Cambridge, Massachusetts, **2022**.
2. Feynman, R., Leighton, R., Sands, M. *The Feynman Lectures on Physics*, Addison Wesley, Boston, Massachusetts, **2005**.

3. Hurwicz, L. *Optimality Criteria for Decision Making Under Ignorance*, Cowles Commission Discussion Paper, Statistics, No. 370, **1951**.
4. Kreinovich, V. Decision making under interval uncertainty (and beyond). In: Guo, P., Pedrycz, W. (eds.), *Human-Centric Decision-Making Models for Social Sciences*, Springer Verlag, **2014**; pp. 163–193.
5. Luce, R.D., Raiffa, R. *Games and Decisions: Introduction and Critical Survey*, Dover: New York, **1989**.
6. Nguyen, H.T., Kreinovich, V. *Applications of Continuous Mathematics to Computer Science*, Kluwer: Dordrecht, **1997**.
7. Odrzywolek, A. *All elementary functions from a single operator*, arXiv:2603.21852v2, April **2026**.
8. Odrzywolek, A. *Diversity of EML-type operators*, arXiv:2609.11210, September **2026**.
9. Thorne, K.S., Blandford, R.D. *Modern Classical Physics: Optics, Fluids, Plasmas, Elasticity, Relativity, and Statistical Physics*, Princeton University Press: Princeton, New Jersey, **2021**.