Measuring Regulatory Complexity*

Jean-Edouard Colliard†
HEC Paris

Co-Pierre Georg‡
University of Cape Town
Deutsche Bundesbank

Abstract

Despite a heated debate on the perceived increasing complexity of financial regulation, there is no available measure of regulatory complexity other than the mere length of regulatory documents. To fill this gap, we propose to apply simple measures from the computer science literature by treating regulation as if it was an algorithm—a fixed set of rules that determine how an input (e.g., a bank balance sheet) leads to an output (a regulatory decision). We apply our measures both to the stylized regulation of a bank in a theoretical model and to actual regulatory texts. Our results emphasize that shorter regulations are not necessarily less complex, as they can also use more "high-level" language and concepts. Finally, we propose a protocol to validate our measures experimentally.

Keywords: Financial Regulation, Capital Regulation, Regulatory Complexity, Basel Accords.

JEL classification: G18, G28, G41.

*We are grateful to participants to the 2016 "Journée de la Chaire ACPR", seminar audiences at Autorité des Marchés Financiers, the University of Bonn and the Bank of England for helpful comments and suggestions. This work was supported by a grant from Institut Louis Bachelier. The views expressed in this paper do not necessarily reflect the views of Deutsche Bundesbank, the ECB, or the ESCB.

†E-Mail: colliard@hec.fr

‡E-Mail: co-pierre.georg@bundesbank.de

1 Introduction

The regulatory overhaul that followed the great financial crisis has triggered a hefty debate about the complexity of financial regulation. For instance, Haldane and Madouros (2012) articulate the view that bank capital regulation has become so complex as to be counter-productive and likely to favor regulatory arbitrage. The Basel Committee on Banking Supervision itself is aware of the issue, and considers simplicity as a desirable objective, to be traded off against the precision of regulation (Basel Committee on Banking Supervision (2013)). In the United States, similar concerns have led to a proposal to exempt small banks from some rules provided that they appear sufficiently capitalized (see Calomiris (2018) for a discussion).

While there is a widespread concern that regulation has become too complex, "regulatory complexity" remains an elusive concept to quantify. An often-used measure is the length of regulation. For instance, Haldane and Madouros (2012) use the number of pages of the different Basel Accords (from 30 pages for Basel I in 1988 to more than 600 pages for Basel III in 2014). While illustrative, such a measure is quite crude and difficult to interpret. For instance, should one control for the fact that Basel III deals with a significantly higher number of issues than Basel I? Is a longer, but more self-contained regulation more complex, or simpler? To guide us through such questions, we lack a framework to think about what complexity means in this context and how it can be measured.

The core idea of this paper is to analyze regulations as algorithms and adapt definitions and measures of algorithmic complexity developed in computer science to the context of financial regulation. Indeed, our starting observation is that a regulation can be seen as a list of instructions and operations that are applied to a set of financial institutions and return a regulatory action (e.g., a sanction). In other words, regulation can be seen as a program that takes financial institutions as inputs and returns a regulatory action as an output. We propose to associate "regulatory complexity" to the complexity of this set of rules, that is, to the complexity of the associated algorithm.

Among the many measures of algorithmic complexity that have been proposed in the computer science literature, we focus in this paper on the "Halstead measures", pioneered by Halstead (1977). As we detail in Section 2, these measures rely on a count of the number of "operators" (e.g., +, −, logical connectors...) and "operands" (variables, parameters...) in an algorithm, and the measures of complexity aim at capturing the number of operations and the number of operands used in those operations. As we will show below, in the context of regulation these measures can help capturing the number of different rules ("operations") in a regulation, whether these rules are repetitive or different, whether they apply to different economic entities or to the same ones, etc.

Our choice of the Halstead measures is motivated by several factors. First, these measures aim at capturing the "psychological complexity" of an algorithm, i.e., how difficult it is to understand, which we feel is what the debate on regulatory complexity is mostly about.[1] Second, these measures are simple and transparent, and thus well-designed for a "proof of concept" study showing that applying measures of algorithmic complexity to financial regulation is potentially fruitful. Third, due to their simplicity the computation of these measures can, to some extent, be automated and generalized to many regulatory texts, so that our approach can easily be replicated and used by other researchers.

To show the potential of this approach, the paper develops several applications of the methodology. First, we develop a simple model in which a regulator designs a capital regulation relying on risk buckets, as in Basel I. We can use our measures to compute the complexity of the regulation chosen. We then study the trade-off for the regulator between achieving a more precise regulation and reducing regulatory complexity, which determines the optimal number of risk-buckets and thus the complexity of the optimal regulation. This example shows that our measure can be used in normative models of regulation. For instance, this allows—in the context of a model—to study whether a complex regulation achieving the first-best is indeed more desirable than a simpler one that still achieves a high level of welfare.

Second, we consider the design of risk weights in the Basel I Accords and measure their complexity. This is a nice testing ground because this part of the regulation is very close to being an actual algorithm. We compare two different methods: (i) We write pseudo com-

[1] As we discuss in Section 2, there are other dimensions of algorithmic complexity that one could also apply to the study of regulation to capture other dimensions, such as the difficulty to implement a regulation.

puter code corresponding to the instructions of Basel I and measure the algorithmic complexity of this code. That is, we use the Halstead measures of algorithmic complexity literally; (ii) We analyze the actual text of the regulation and classify words according to whether they correspond to what in an algorithm would be an operand or an operator, and compute the same measures, this time trying to adapt them from the realm of computer science to a plain English text. The measures we obtain using both approaches are highly correlated, from which we conclude that our measures can be used without actually "translating" a regulatory text into a computer code, which is of course a time-consuming task, as they can be proxied by studying the text directly.

Third, we apply our text analysis approach to the different titles of the 2010 Dodd-Frank Wall Street Reform and Consumer Protection Act and give some descriptive results on which titles are more complex according to different dimensions. In particular, we note that some titles have approximately the same length and yet differ significantly along other measures, which shows that our measures capture something different from the mere length of a text. Because the Dodd-Frank Act covers many different aspects of financial regulation, when doing this analysis, we created a large dictionary of operands and operators in financial regulation, which we plan to make available to researchers interested in using these measures on different texts.

Finally, we plan to provide some experimental validation of the validity of our methodology. There is a literature in computer science studying whether different measures of algorithmic complexity correlate with mistakes made by the programmers or the time they need to code the program (e.g., Canfora *et al.* (2005)). In the same spirit, we plan to ask students to compute regulatory ratios using different Basel-I type rules with different levels of complexity. We want to test whether our measures are good predictors of mistakes made by the students and/or of the time they need to perform the computation, whether this prediction depends on the student's background and training, etc. In this preliminary version, we only outline an experimental protocol and leave the conduct of the experiment for future research.

This paper is part of a growing literature concerned with the market or government failures created by complexity, in particular "psychological complexity", i.e., the difficulty for agents to understand a product, contract, or rule. Hakenes and Schnabel (2012) develop a model of "capture by sophistication" (Hellwig (2010)) in which some regulators cannot understand complex arguments and "rubber-stamp" some claims made by the industry so as not to reveal their lack of sophistication. Asriyan *et al.* (2018) propose a diametrally opposed theory in which regulatory complexity obtains in a political economy setting when policymakers are more informed about which regulations are necessary and public trust of the policymaker is high. Empirical measures of complexity would be necessary to test which theory better explains the data. Rochet (2010) is concerned that regulatory complexity makes regulation opaque to outsiders, so that regulators can become captured by the industry without any external checks and balances. In a broader context, some papers also study how sophisticated agents can strategically exploit complexity to increase their market power (e.g., Carlin (2009)). Finally, Arora *et al.* (2009) argue that computational complexity creates a form of asymmetric information problem, an example being the pricing of some derivatives.

Our paper is also related to a literature in behavioral economics that models economics agents as computer programs. Rubinstein (1986) studies repeated games played by Turing machines. The complexity of a player's strategy is measured by the number of states that enter the machine, and the outcome of repeated games for instance can change dramatically if players prefer less complex strategies, all else equal, even by an infinitesimal amount. While the aim and context of this literature are very different, we share the analogy between economic behaviors and algorithms and the use of an algorithmic measure of complexity.

There is also a growing empirical literature proposing different measures for the complexity of various economic objects, such as financial products (Célérier and Vallée (2017)) and prices (see Ellison (2016) for a survey of the literature on "obfuscation"). There is also a related literature in law proposing to measure the complexity of legal texts through the number of references to other legal texts and the position of a particular law in the associated network (e.g., Li *et al.* (2015)). While all these measures are interesting and complementary, we believe our approach relying on algorithmic complexity is new and particularly well suited to the study of regulatory complexity.

Finally, there is a large literature in computer science proposing different measures of algorithmic complexity, whose application to regulatory complexity could also be considered in future research. Another very popular measure in this literature is for instance the "cyclomatic complexity" of McCabe (1976). We refer the interested reader to Yu and Zhou (2010) for a recent survey.

2 Framework

2.1 General definitions

Different authors, policymakers and industry participants have different concepts in mind when referring to "regulatory complexity", because the term "complexity" is somewhat vague. In this section we introduce preliminary definitions so as to clarify the different dimensions of complexity, and introduce the ones we are going to measure in this paper.

We start by making the analogy between regulations and algorithms more precise. The goal of an algorithm is to solve a "problem" or a "computation", which in general can be seen as associating the right "outputs" to elements in a set of "inputs". In the case of regulation, the "input" is a regulated entity (e.g., a bank and its balance sheet, characteristics about its operations, etc.), and the output a regulatory action (letting the bank operate, imposing a fine, etc.). Formally, we define:

Definition 1. *A regulatory problem is a mapping $f : \mathscr{E} \to \Sigma$ from the set of regulated entities $\mathscr{E}$ to a set of regulatory actions Σ.*

An algorithm is a set of mechanical rules, such that by following them we can compute $f(x)$ given any input x. Similarly, a regulation is a set of rules that implement the right regulatory action to any regulated entity:

Definition 2. *A regulation $\tilde{f}$ is a list of elements taken in a vocabulary $\mathscr{V}$. This list of elements is interpreted through a language, and implements f.*

It is important here to notice that, in the same way that different algorithms can solve the same problem, different regulations $\tilde{f}$ can solve the same regulatory problem f. To put it simply, there are many different ways of writing the Basel I Accords that would all lead to the same computations of capital requirements for any possible bank balance sheet.

Finally, once a particular algorithm to solve a problem has been chosen, the last step is to actually run the algorithm, which may take more or less time and computing power. Similarly, following the rules set in a given regulation may be more or less complicated for the regulatory authority. We call this last step "supervision":

Definition 3. *Supervision is the act of following $\tilde{f}$ to evaluate $f(e)$ for a given entity $e \in \mathscr{E}$.*

We can now give defining properties of measures of regulatory complexity corresponding to different dimensions. Assume that we have a set $\tilde{\mathscr{F}} = \{\tilde{f}_1, \tilde{f}_2 ..., \tilde{f}_n\}$ of regulations solving the same regulatory problem f, and a set $\mathscr{E} = \{e_1, e_2, ..., e_m\}$ of regulated entities. Elements of these sets could be empirically observed (actual regulatory texts, actual banks) or hypothetical (variants on the text, hypothetical banks). Following the previous definitions, we can define a measure of regulatory complexity and give necessary conditions for different types of measures as follows:

Definition 4. *A measure of regulatory complexity μ is a mapping $\mu : \tilde{\mathscr{F}} \times \mathscr{E} \to \mathbb{R}$. Then:*

1. *If μ is a measure of problem complexity, then $\mu(\tilde{f}, e)$ is constant in $\tilde{f}$ and e.*
2. *If μ is a measure of psychological complexity, then $\mu(\tilde{f}, e)$ is constant in e but not necessarily in $\tilde{f}$.*
3. *If μ is a measure of computational complexity, then $\mu(\tilde{f}, e)$ may depend both on e and $\tilde{f}$.*

These properties characterize an important distinction between three forms of regulatory complexity:

(i) Regulatory complexity may mean that the regulatory problem is complex, e.g., it deals with many different aspects of a bank's business, foresees a large number of regulatory actions, etc. We call this dimension the *problem complexity* of regulation. Problem complexity depends on f, but is independent of which regulation $\tilde{f}$ implements f.

(ii) Regulatory complexity may also mean that the actual regulation used to solve the regulatory problem is complex, which may be due both to the complexity of the problem f and to the complexity of the particular $\tilde{f}$ that solves the problem. Following the computer science literature, we call this dimension the *psychological complexity* of regulation, as it reflects the difficulty of understanding a particular solution to a problem.

(iii) Finally, regulatory complexity may mean that applying a regulation to a particular entity or group of entities is costly in terms of time and resources. The cost can be incurred by the supervisor (supervision costs) and by the regulated entities (compliance costs). Imagine for instance a regulation that exempts small banks from most rules. It could then be the case that the regulatory text is complex, that applying it to large banks is costly, but that applying it to small banks is simple. Thus, this dimension depends on the entity to which the regulation is applied. Following again the computer science literature, we call this dimension the *computational complexity* of regulation.

Example: Length of bank capital regulation. In the example of capital regulation, a regulated entity is a bank, represented for instance by a list B of balance sheet items and values. The regulatory problem is to associate any possible bank balance sheet B to an action, the simplest ones being for instance "pass" or "fail", i.e. $\Sigma = \{0, 1\}$. Regulation is then a series of operations on balance sheet items that ends with an outcome $\sigma \in \Sigma$.

Haldane and Madouros (2012), for instance, measure the complexity of banking regulation by the number of pages of the different Basel Accords. In our framework, the exact text of the Basel Accords is a particular regulation $\tilde{f}$ to solve an underlying regulatory problem. The length of the text is a particular measure. Clearly, this measure depends on how the text is written, but not on which bank we apply the regulation to. In our framework, length is thus a measure of psychological complexity, but not of problem complexity.

2.2 Halstead Measures

We now develop particular measures of complexity by adapting the work of Halstead (1977). Since we apply these measures to regulatory texts and not to data on regulated entities, our aim here is to measure problem complexity and psychological complexity, but not computational complexity.

In order to apply this approach, we need to consider a regulation $\tilde{f}$ as an (ordered) list of "words" (elements in a language) $\tilde{f} = \{w_1, w_2...w_N\}$, in which we can classify the w_i into two lists: a list of N_1 operators and a list of N_2 operands, with $N_1 + N_2 = N$. We also define $\mathcal{O} = \{o_1, o_2...o_{\eta_1}\}$ and $\Omega = \{\omega_1, \omega_2...\omega_{\eta_2}\}$, the sets of all operators and operands that appear in $\tilde{f}$. η_1 is the total number of unique operators, and η_2 the total number of unique operands.

Using Halstead's definitions, operands in an algorithm are "variables or constants" and operators are "symbols or combinations of symbols that affect the value or ordering of an operand". Consider for instance the following "pseudo-code" to compute the absolute value of a number:

$$\text{if } x \geq 0 \quad , \quad y = x$$
$$\text{if } x < 0 \quad , \quad y = -x$$

In this code, the operators are if, $\geq, <, =, -$, and the operands are $x, y, 0$. We have $\eta_1 = 5, N_1 = 7, \eta_2 = 3, N_2 = 8$. A simple measure of complexity, corresponding to the length, is simply the total number of operators and operands, called the *volume*:[2]

Definition 5. *The volume V of regulation $\tilde{f}$ is equal to $N_1 + N_2$.*

The volume is a (simple) measure of psychological complexity. How can one obtain a measure of problem complexity, without knowing all the possible $\tilde{f}$ that implement f? Halstead's answer is to look at the theoretically shortest program that can solve the problem, in the best possible programming language. Defining this algorithm is actually simple. Going back to our example of the absolute value, the shortest possible program is:

$$y = abs(x)$$

This is the shortest because any program to compute the absolute value of a number would need to specify the input, the output, an assignment rule, and an operation (here, an

[2]The approach in Halstead (1977) is slightly more complicated than what we present here, as Halstead wants a measure that does not depend on the alphabet used to code the program. We abstract from this problem, which we don't think is first-order in the context of regulation.

operation that already exists in the programming language). More generally, for any problem, the shortest program would still contain a minimum number of operands η_2^* that represent the number of inputs and outputs of the program. All the operations transforming the inputs into outputs would already be part of the language as a single built-in function. The number of operators is then $\eta_1^* = 2$, and the number of operands η_2^*. The volume of this minimal program, called *potential volume*, is thus:

Definition 6. *The potential volume V^* of $\tilde{f}$ is equal to $2+\eta_2^*$.*

Importantly, if one assumes that the list of inputs and outputs never includes some unnecessary ones, V^* will be independent of $\tilde{f}$. That is, V^* is a measure of problem complexity.

An interesting question to ask is whether an algorithm is close to the shortest algorithm or not. Adapting Halstead (1977), we define the *level* of an algorithm as:

Definition 7. *The level L of $\tilde{f}$ is equal to $\frac{V^*}{V} = \frac{2+\eta_2^*}{N_1+N_2}$.*

To better understand what the level captures, we can write:

$$\frac{1}{L} = \frac{\eta_1+\eta_2}{2+\eta_2^*} \times \frac{N_1+N_2}{\eta_1+\eta_2}. \tag{1}$$

The first term in this product reflects the number of operations performed instead of using a "built-in" operation, and the number of unnecessary operands that are introduced (e.g., intermediary results). The second term is simply the average number of repetitions of the same elements in the program.

We think the measure L has a nice interpretation in the context of regulatory complexity. If L is high (close to 1) this means that the regulation has a very specific vocabulary, a technical jargon, that is opaque to outsiders. Conversely, a low value of V means that the regulation starts from elementary concepts and operations. In particular, a low value of V means that η_1 is greater than 2, so that the representation of regulation defines auxiliary functions (operators) in terms of more elementary ones.

Under this interpretation, we can see that there is a very intuitive trade-off between volume and level. One can make the regulation shorter by using a more specialized vocabulary, but this is going to increase the level and make the regulation more opaque. Conversely, one can make regulation more accessible or self-contained by defining the specialized words in terms of more elementary ones, but the cost is a greater length. To capture this trade-off, we assume that psychological complexity can be captured by a *cost of complexity* function:

Definition 8. *The cost of complexity $C(V,L)$ is increasing in both V and L.*

In particular, in the following applications we will compute both V and L and illustrate that it can be informative to compute the level as a second dimension of complexity on top of the volume, or length, of the regulation.

3 A model of coarse capital requirements

In this section we introduce a very simple model of bank capital requirements in which we use the Halstead approach to measure the complexity of regulation. The outcome of the model is an optimally coarse capital regulation relying on a finite number of risk buckets, whose number depends negatively on the cost of complexity.

3.1 The banking model

Consider a bank with 1 in assets, that can be financed either with deposits D or equity E. In case the bank fails, depositors are reimbursed by the government using public funds, which have a marginal cost of $1+\lambda$. These losses can be mitigated by asking the bank to use more equity, but we take as given that equity has a marginal social cost of $1+\delta$.

There is a continuum $x \in [0,1]$ of asset types. The bank starts with an asset of type x, drawn from the uniform distribution over $[0,1]$. With probability p, the economy is growing and asset x pays $r(x)$. With probability $1-p$, the economy enters a recession and the asset pays only $1-x$, i.e., the bank makes a loss of x on its investment. If $E<x$ the bank defaults, and the government has to repay $D-(1-x)=x-E$ to the depositors.

For a given level of equity E and an asset type x, total welfare writes as:

$$pr(x)+(1-p)[1-x-\lambda\min(x-E,0)]-\delta E. \tag{2}$$

We want to derive a regulation that maximizes total welfare. As $pr(x)+(1-p)(1-x)$ is exogenously given, we can consider the following objective function:

$$\mathscr{W}(E,x) = -\lambda(1-p)\min(x-E,0) - \delta E. \tag{3}$$

As long as $E < x$, we have $\partial W/\partial E = \lambda(1-p)-\delta$. We assume this quantity to be positive: the social cost of capital is lower than the expected gain of reducing losses to the public sector. It is then clear that the optimal regulation would be to have $E^*(x) = x$ for any x, so that the bank never defaults. Total expected welfare would then be:

$$\int_0^1 \mathscr{W}(x,x)dx = \int_0^1 -\delta x dx = -\frac{\delta}{2}. \tag{4}$$

Such a regulation requires to associate a continuum of different asset types to different levels of capital, which may be very complex, and hence costly.

We assume instead that the regulator defines different buckets, that is, intervals $[a_i, b_i]$ such that if $x \in [a_i, b_i]$ then $E \geq E_i$. As we show in the Appendix A, for a given interval $[a,b]$ the optimal capital requirement $E^*_{a,b}$ is given by:

$$E^*_{a,b} = b - \delta\frac{b-a}{\lambda(1-p)}. \tag{5}$$

Note that we indeed have $a \leq E^*_{a,b} \leq b$. This means that banks with assets x close to a will be over-capitalized (they have more capital than what is necessary to sustain the losses x), while banks with assets x close to b will be undercapitalized (they default with probability $1-p$).

We obtain that the optimal welfare over interval $[a,b]$ is given by:

$$W_{a,b}(E^*_{a,b}) = \delta(b-a)\left[\frac{\delta(b-a)}{2\lambda(1-p)} - b\right]. \tag{6}$$

Using this expression, we can determine the optimal intervals chosen by the regulator. As shown in the Appendix, if the regulator uses I intervals it is actually optimal to split $[0,1]$ into

I intervals of equal length, and we compute that total welfare is given by:

$$\mathscr{W}(I) = \sum_{i=0}^{I-1} W_{i/I,(i+1)/I}(E^*_{i/I,(i+1)/I}) = -\frac{\delta}{2} - \frac{\delta}{2I\lambda(1-p)}[\lambda(1-p)-\delta]. \tag{7}$$

Total welfare is thus increasing in I, and converges to the continuous case $-\delta/2$ as $I \to +\infty$. Without any cost of complexity, it would be optimal to define as many risk buckets as possible.

3.2 Complexity of the risk-buckets

Let us now estimate the complexity of the regulation for a given number I of intervals. In general, such a regulation can be written as follows:

- if $x \geq 0$ and $x < \bar{x}_1$ then $E \geq E_1^*$
- if $x \geq \bar{x}_1$ and $x < \bar{x}_2$ then $E \geq E_2^*$
- ...
- if $x \geq \bar{x}_{I-1}$ then $E \geq E_I^*$

This regulation uses the following operands: x ($2I-1$ times), 0 (once), $\bar{x}_1, \bar{x}_2, \bar{x}_{I-1}$ (twice each), E (I times), and $E_1^*, E_2^* ... E_I^*$ (once each). The total number of operands is thus $N_2 = 6I-2$, the total number of unique operands $\eta_2 = 2I+2$. For the operators, we have “if” (I times), “≥” ($2I$ times), “≤” ($I-1$ times), and “then” (I times). This gives us $N_1 = 5I-1$ operators in total, and $\eta_1 = 4$ unique operators.

The total volume of the regulation is thus $V(I) = N_1 + N_2 = 11I-3$ and increases linearly in I. The potential volume is $V^* = 2+\eta_2^* = 2I+4$, and the level is:

$$L(I) = \frac{V^*}{V} = \frac{2I+4}{11I-3}. \tag{8}$$

In particular, the level is decreasing in I. It can also be decomposed as:

$$L = \frac{2+\eta_2}{\eta_1+\eta_2} \times \frac{\eta_1+\eta_2}{N_1+N_2} = \frac{2I+4}{2I+6} \times \frac{2I+6}{11I-3}. \tag{9}$$

The first term measures the drop in level due to relying on basic operators instead of having a built-in function. This ratio increases in I. The second term in the inverse of the number of repetitions in the program and decreases in I. Thus, L decreases in I due to the fact that its structure is very repetitive. Figure 1 illustrates these facts by plotting the volume and the level as functions of I.

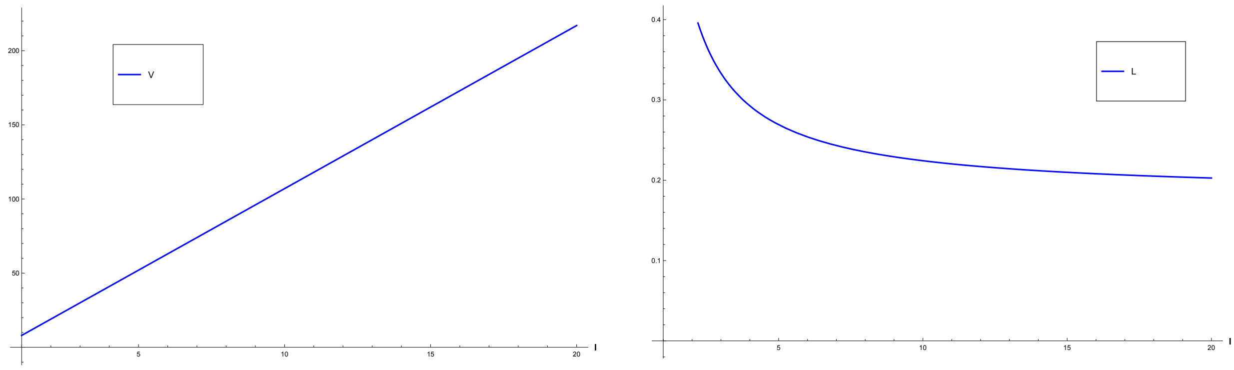

Figure 1: Volume and Level as a function of I.

3.3 Optimal Regulation

The optimal regulation should take into account both the impact on economic welfare and the cost of regulatory complexity. The optimal number of risk buckets is given by:

$$I^* = arg \max_I \mathscr{W}(I) - C(V(I), L(I)). \tag{10}$$

The first-order condition can be expressed as:

$$\frac{\delta[\lambda(1-p)-\delta]}{2\lambda(1-p)} \times \frac{1}{I^{*2}} - 11C_1(V(I^*), L(I^*)) + \frac{50}{(11I^*-3)^2} C_2(V(I^*), L(I^*)) = 0. \tag{11}$$

In particular, as C_1 and C_2 are assumed to be positive, unless C_1 converges to 0 when V goes to infinity the optimal number of intervals is finite. In other words, a coarse regulation is optimal because it reduces complexity, despite being less efficient from an economic perspective. The optimal number of intervals I^* results from a trade-off between increasing welfare, reducing volume (which increases in I), and reducing level (which decreases in I). In particular, depending on the shape of the cost function it is possible that increasing I

actually decreases complexity for some values of I.

4 Basel I

We now apply our measure empirically to an actual text, the 1988 Basel I Accords (Basel Committee on Banking Supervision (1988)). We focus on Annex 2, "Risk weights by category of on-balance-sheet asset". As we will illustrate below, this is a natural starting point because this part of the regulation can easily be described as an algorithm. This allows us to compute our measures based both on an algorithmic representation of Basel I and on the actual text. We then compare the results obtained in both cases and conclude that the text-based method is a good proxy for the more literal application of measures of algorithmic complexity.

4.1 Basel I as an algorithm

The Basel I Accords are a 30-page long text specifying how to compute a bank's capital ratio. This is done by mapping different asset classes to different risk buckets, and different capital instruments to different weights. The regulation then compares the risk-weighted sum of assets to the weighted sum of capital, and the ratio has to be higher than 8%. As this succinct description makes clear, Basel I is easily described as an algorithm. We actually wrote "pseudo-code" that implements the computation of risk-weighted assets described in the Annex 2 of the text, i.e., our code maps a bank balance sheet to total risk-weighted assets under Basel I. We give this program in Appendix B. In this section, we briefly explain the structure of the code and give the measures based on the program.

Annex 2 of the Basel I text is a list of balance sheet items associated with different risk weights (5 different risk-weights in total). For instance, in the 20% risk-weight category we have "Claims on banks incorporated in the OECD and loans guaranteed by OECD incorporated banks". In our code this is translated into:

```
if ((ASSET_CLASS == "claims" and ISSUER == "bank" and ISSUER_COUNTRY == "oecd")
  or (ASSET_CLASS == "loans" and GUARANTOR == "bank" and GUARANTOR_COUNTRY == "oecd")
```

```
) then:
  risk_weight = 0.2;
```

We can easily identify the operands and operators in such a piece of code, and compute our measures of complexity. The operands are the different asset classes (e.g., `ASSET_CLASS`, `claims`), attributes (e.g., `ISSUER_COUNTRY`, `GUARANTOR`), values of those attributes (e.g., `oecd`, `bank`), and risk-weights (e.g., `risk_weight`, `0.2`). The operators are `if`, `and`, `or`, `else`, `==`, `>`, `<=`, and `!=`. Given our algorithmic representation of Basel I, we find that $N_1 = 172, N_2 = 184, \eta_1 = 8, \eta_2 = 45$.

These numbers are by themselves not very interesting, as we have no benchmark to compare them to. We can go further by computing how much the regulation of different asset classes contributes to the total. In Table 1, we report the values of V, V^*, and L for each of the 19 items in covered by Basel I, as well as the total. Moreover, we compute the "marginal contribution to level" of each item by computing the (relative) difference between the actual total level and what total level would be if we took out the part of the algorithm dealing with this item.

In particular, the table reveals that the different items are very hetreogeneous in terms of their contributions to the total level. If we look at the extreme cases, excluding item 3d for instance would reduce total level by 12%, whereas excluding item 4a would increase the total level by 6%. Interestingly, the correlation between total volume and the contribution to level is extremely small (-0.03). Thus, what the contribution to level captures is not that longer items reduce the total level of regulation, but rather that some items mostly use operands and operators that are also used elsewhere in the text, whereas other items introduce many new terms (in particular, item 3d).

Finally, one last interesting exercise is to compute how the different measures evolve as the code considers additional items. In order to do this, we compute our measures for an hypothetical regulation with only the first asset class, then the first two, the first, three, etc. Figure 2 below plots how V, V^*, and L evolve.

As can be seen on the figure, each new item increases volume by 21 units on average, but potential volume only by 2. As a result, the level of regulation decreases with almost every

Table 1: Complexity measures for the items in Appendix 2 of Basel I - Algorithm.

Item	V	V^*	L	Contribution to level (in %)
1a	4	4	1.00	-1.14
1b	19	10	0.53	-5.64
1c	15	9	0.60	-4.40
1d	11	7	0.64	-3.19
2a	40	12	0.30	-12.66
3a	64	12	0.19	-8.95
3b	23	11	0.48	-6.91
3c	31	13	0.42	-9.54
3d	39	13	0.33	-12.30
3e	7	6	0.86	2.33
4a	16	9	0.56	6.43
5a	8	6	0.75	-0.12
5b	15	10	0.67	-4.40
5c	19	11	0.58	-5.64
5d	11	8	0.73	3.40
5e	15	7	0.47	4.49
5f	7	5	0.71	2.33
5g	11	8	0.73	3.40
5h	1	2	2.00	-0.28
Total	356	47	0.13	

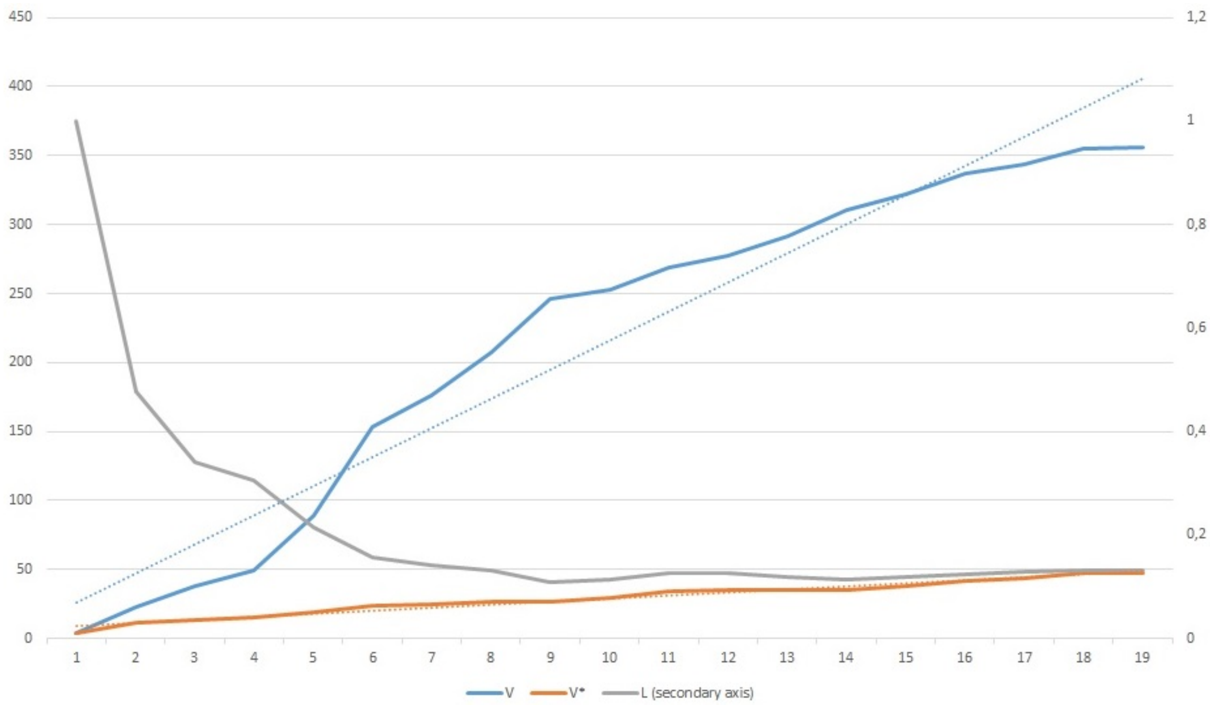

Figure 2: Incremental complexity measures for the items in Appendix 2 of Basel I.

new item. This figure illustrates the repetitive nature of the Basel I definition of risk-weights.

4.2 A textual analysis of Basel I

We now repeat the same analysis of the Appendix 2 of Basel I, but relying this time on the actual text and not on our “translation” into code. We want to classify as “operands" the words that have the same function as operands in the program, and similarly for operators. This is not a completely trivial task and there is some judgement involved, as the logic of the text in plain English and the logic of the algorithm are a bit different. In particular, the text leaves some elements implicit, whereas the algorithm has to be explicit about all the steps of the computation.

We classify as operators all the words or combinations of words that correspond to operations or logical connections, such as “and” or “excluding”. Operands are all the words that correspond to economic entities (e.g., “bank” or “OECD”), concepts (e.g., “maturity” or “counterparty”), and values (e.g., “one year"). Using this approach, we classify 72 unique words out of the 86 words vocabulary used by the text. The remaining words are used for grammatical reasons and do not really correspond to operands or operators (e.g., “by”, “on”, “the”, etc.), hence we don’t take them into account. Table 2 gives the top 10 operands and operators that we identify in the text.

Table 2: Top 10 words in each category, Appendix 2 of Basel I.

Operands		Operators	
claims	15	and	12
banks	10	other	6
OECD	10	or	4
central	9	outside	4
guaranteed	6	excluding	2
incorporated	5	non	2
currency	4	unless	2
entities	4	up to	2
governments	4	above	1
sector	4	all	1

We then reproduce Table 1 using the measures based on our text analysis. The patterns are quite similar to those observed in the algorithmic version. We next turn to comparing the two approaches more systematically.

Table 3: Complexity measures for the items in Appendix 2 of Basel I - Text analysis.

Item	V	V^*	L	Contribution to level (in %)
1a	1	3	3.00	-0.40
1b	16	13	0.81	-6.87
1c	9	8	0.89	-3.75
1d	15	13	0.87	-6.41
2a	15	14	0.93	-6.41
3a	22	18	0.82	0.41
3b	14	11	0.79	-5.96
3c	34	18	0.53	-14.29
3d	17	15	0.88	-7.33
3e	5	7	1.40	1.98
4a	21	18	0.86	6.60
5a	5	7	1.40	-0.71
5b	14	14	1.00	-5.96
5c	19	15	0.79	-6.84
5d	9	11	1.22	-1.02
5e	8	7	0.88	2.12
5f	12	9	0.75	1.85
5g	10	9	0.90	1.30
5h	3	3	1.00	-1.22
Total	283	83	0.29	

4.3 Comparison

To compare the measures obtained with the algorithmic approach and the text analysis, we compute the correlation between the values of V, V^*, L, and contribution to level in the two cases. Moreover, since each item itself is not a functioning code (for instance the text of item 1a is only one word, “Cash”), the level sometimes returns a value above 1, which is in principle not possible. Thus, we also introduced the “capped level”, which is the minimum of 1 and the level. Table 4 gives the correlation coefficients for the different measures, as well as the Spearman rank correlation coefficients.

The correlation coefficients we obtain are quite large, which shows that the text-based

Table 4: Correlation coefficients and Spearman rank correlation coefficients between the measures based on the algorithm and the measures based on the text.

	Correlation	Rank correlation
V	0.64	0.83
V^*	0.85	0.83
L	0.33	0.62
Contribution to level	0.79	0.79
Capped L	0.53	0.62

analysis and the algorithm-based analysis are capturing similar patterns, especially if one focuses on ordinal comparisons. The highest correlation we obtain is for V^*, which is natural since it relies on counting the unique economic concepts used by the regulation, which have to be more or less the same in the text and in the code (the difference coming from the fact that the code needs to be a bit more explicit). The volume is less correlated, and as a result the level (equal to V/V^*) shows a lower correlation, but still a high rank correlation of 0.62. Focusing on the contribution to total level instead of the level itself seems to improve the correlation between the two measures, presumably because implicitly this measure neutralizes the difference of "styles" between the text and the code.

Overall, we conclude from this comparison that measures of regulatory complexity relying on a text analysis can be a good proxy for the more theoretically founded measures based on the algorithmic version, especially if one focuses on ordinal comparisons and on volume, potential volume, and contribution to total level. Given this result, we now apply the text-based approach to a more comprehensive regulatory text.

5 Complexity of the Dodd-Frank Act

5.1 Methodological issues

One of the benefits of the Halstead measures as implemented by the text analysis approach of Section 4.2 is that they can be applied automatically to a regulatory text, without having to first "translate" the text into a proper algorithm or to analyze the text manually. The only thing that is needed is a vast dictionary of regulatory terms, with a classification of words into operators and operands.

To start building such a dictionary and illustrate the applicability of our approach, we compute our complexity measures for the different titles of the 2010 Dodd-Frank Act. There are two reasons for this choice. First, the Dodd-Frank Act is one of the key regulations introduced after the financial crisis, which has triggered a lot of debates, in particular regarding its perceived complexity. Second, this text touches upon a wide range of issues in finance, so that by classifying the words of the Dodd-Frank Act we hope to create a relatively complete dictionary that can be used for many other regulatory texts.

There are also some drawbacks of using the Dodd-Frank Act as an example. First, the Act uses a lot of external references. As an example, Section 201 (5) reads as follows:

(5) COMPANY. - The term "company" has the same meaning as in section 2(b) of the Bank Holding Company Act of 1956 (12 U.S.C. 1841(b)), except that such term includes any company described in paragraph (11), the majority of the securities of which are owned by the United States or any State.

How should one deal with such a case? A possible solution would be to include the text referenced in the example as being implicity part of the Act. However, with such an approach we would quickly run into the "dictionary paradox" (every reference refers to other texts). Instead, and more consistent with the Halstead approach, we consider that if a legal reference is mentioned it is part of the "vocabulary" one has to master in order to read the Act. It is not clear however if such references should be counted as operands or operators. We chose not to count them in our measures, but including them does not qualitatively change the results we report below.

A second issue is that the Dodd-Frank Act is a high-level text which in many instances mandates regulatory agencies to draft more precise regulations, or sets how to organize those regulatory agencies. Title X for instance sets up the Bureau of Consumer Financial Protection (BCFP) and the rules it must follow. In line with the framework of Section 2, this can be considered as a regulation where the regulated entity is the BCFP itself. However, one must keep in mind that measuring the complexity of the rules organizing the BCFP is of course something different from the complexity of the regulations written or applied by the BCFP.

The text of the Dodd-Frank Act being much longer, richer, and less close to an algorithm than the Appendix 2 of Basel I, we need to refine our classification of words. We define two categories of operators: (1) *logical operators* are all the words indicating an operation, a condition, a negation, etc. ; (2) *regulatory operators* are words indicating that regulation affects the behavior of a regulated entity. We also define three categories of operands: (1) *economic operands* are all words referring to an economic entity or concept, or an economic action; (2) *attributes* are values given to some economic operands or qualifiers; (3) *legal references* are titles of other laws and regulatory texts. We provide a list of the most frequent words of each type below. Finally, words that cannot be classified include *function words* which mainly have a grammatical function, and *other words*.

5.2 Results

Applying the same approach as in 4.2 to the 16 Titles of the Dodd-Frank Act plus its introduction, we create a dictionary containing: 667 operators (374 logical connectors and 293 regulatory operators), 16,474 operands (12,910 economic operands, 560 attributes, and 3,004 legal references), as well as 711 function words and 291 other, unclassified words (that is, we classified 98.4% of the 18,143 unique words used in the Dodd-Frank Act). Table 5 shows the top 10 words in each category as well as the number of occurrences:

Similarly to what we did in Section 4.2, we now compute different measures for the different titles of the Dodd-Frank Act. As we are particularly interested in comparing volume and level, Figure 5.2 gives a scatter plot showing these two measures for each title. As we see on the graph, there is a negative correlation between level and volume, but these two measures are not perfectly correlated. There are 9 titles with less than 5000 words and different levels, and 9 titles with a level between 0.15 and 0.275 and very different volumes. Thus, these two measures are capturing different dimensions.

It is also instructive to decompose the level as suggested in Section 2.2: the inverse of level is the product of the number of repetitions in the text and the number of extra operators.

Table 5: Top 10 words in each category, entire Dodd-Frank Act.

Operands				Operators			
Economic		Attributes		Regulatory		Logical	
financial	2325	7	5438	shall	3601	or	181077
mission	2191	2	4747	require	1546	not	1963
ban	1493	9	2584	establish	632	including	762
amend	1382	10	968	required	586	provided	465
form	1362	appropriate	829	determine	541	subject to	464
action	1345	15	776	enforce	432	non	448
bank	1244	20	616	prescribe	421	include	361
ring	1172	necessary	311	designated	405	whether	285
use	1170	directly	190	established	287	except	255
date	1116	prudent	181	apply	239	more	248

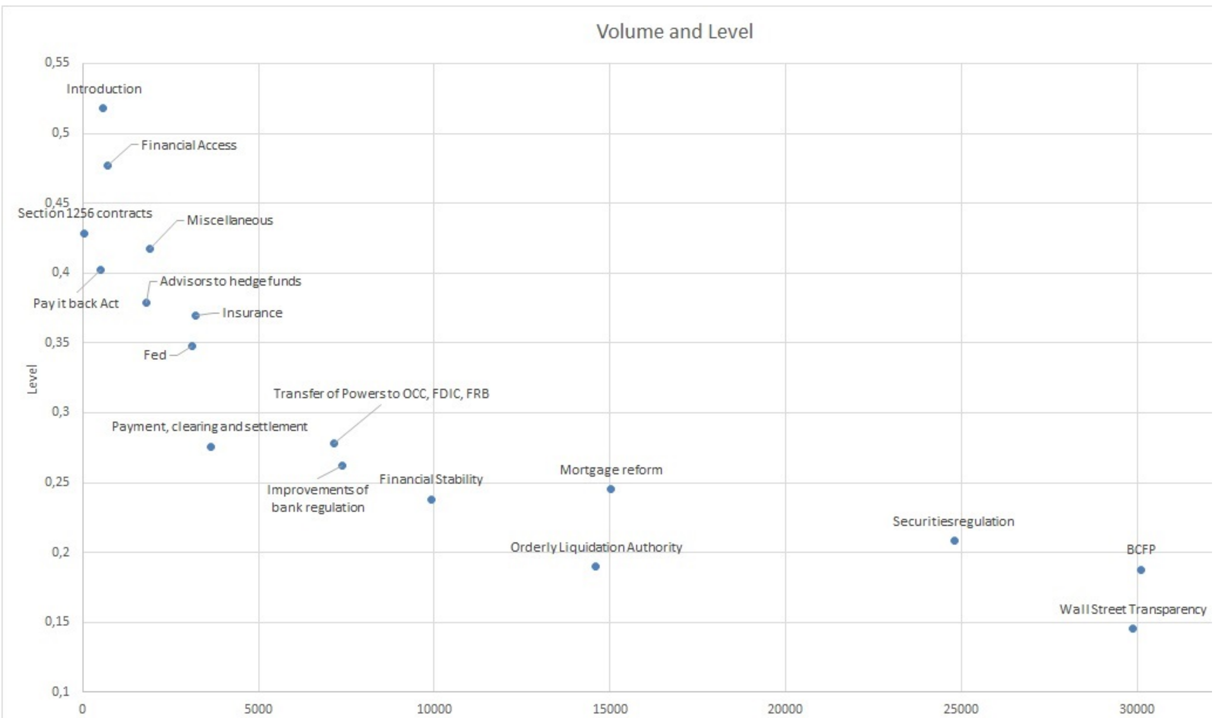

Table 6 below gives the volume of each title, the inverse of level, and the decomposition. The table illustrates that there is very little variation in the number of extra operations, so that the heterogeneity in levels is mostly driven by the number of repetitions.

Table 6: Measures and decomposition of level, titles of the Dodd-Frank Act.

Title number	Volume	$1/Level$	Extra Operations	Repetitions
0	587	1,93	1,08	1,79
1	9905	4,20	1,08	3,88
2	14602	5,26	1,08	4,89
3	7164	3,59	1,07	3,37
4	1793	2,64	1,11	2,38
5	3212	2,70	1,08	2,50
6	7396	3,81	1,08	3,52
7	29882	6,88	1,07	6,46
8	3632	3,62	1,10	3,29
9	24790	4,80	1,06	4,51
10	30112	5,33	1,06	5,04
11	3104	2,88	1,10	2,62
12	709	2,10	1,08	1,95
13	497	2,49	1,10	2,27
14	15022	4,08	1,07	3,82
15	1915	2,39	1,08	2,22
16	49	2,33	1,38	1,69
Total	154371	9,37	1,04	9,01

In addition to these results, in order to perform our analysis we created a dictionary of 18,143 different words, which can be used to compute complexity measures on other regulatory texts. In the future, we plan to make this dictionary available online, as well as the interface we used to classify the words in the first place. We hope that through this collaborative tool other studies of regulatory complexity will be conducted, so that for instance the complexity of different types of regulation or regulations in different countries can be compared.

6 Experiments

6.1 Motivation

The last step of our analysis is to use experiments, with a twofold objective:

First, while our measures are based on a (simple) theory of what regulatory complexity means, they are necessarily somewhat arbitrary and it is necessary to prove that they are indeed useful to capture some dimensions of regulatory complexity. In computer science, complexity measures are tested by asking different programmers to write the same code. It is then easy to check whether the mistakes they make or the time they take to perform the task are correlated with different measures of complexity. In the same vein, we want to ask experimental subjects to compute some regulatory ratios and see if the quality of their output is correlated with our measures of regulatory complexity.

Second, since it is clear that regulatory complexity is a multidimensional object, we want to shed some light on how these different dimensions inetract with each other. Formally, recall that we assumed the cost of complexity to depend on volume and level through a function $C(V, L)$. We would like to know more about the shape of this function, which can be done by estimating the "marginal rate of substitution" between V and L. In particular, some papers in the theoretical literature implicitly assume a Leontief function: agents are able to understand up to a certain level of complexity, and then don't understand at all. We can test this hypothesis. We can also study the heterogeneity of the function C among our experimental participants, and how it is determined for instance by background, education, professional experience, etc.

6.2 Experimental protocol

Our experiments rely on asking subjects to compute a Basel I style capital ratio, based on regulations written as in Section 4. In each experiment, the subject is shown a "regulation" (a set of instructions involving different asset classes, characteristics, risk weights, etc.) and a bank balance sheet. The subject is asked to compute the bank's risk-weighted assets following the instructions. We record the numerical value entered by the subject and the time taken to answer the question.

To generate a set of reasonable instructions of varying complexity, we wrote an algorithm that "randomly" generates regulations akin to those in Basel I. For our random regulation to have the same structure as the Basel I regulation text (see Section 4.1), we decide upfront on the number of `IF-THEN-ELSE` clauses we want to have. As with the actual Basel I regulation, we use 6 clauses in total. Within each clause, the algorithm then selects a number of random conditions (smaller or equal than some fixed positive bound, in our case 10). Each condition consists of operators and operands, e.g. `ASSET_CLASS == ''cash''` that can be combined by `AND` and `OR` statements. We use only operands and operators that also exist in the Basel I regulation. Operands in our random regulation generator can take exactly the values they can take in the original Basel I text. For example, `ASSET_CLASS` can take the values {`cash, claim, loan, premises, plant, equipment, real_estate, other_fixed_assets, other_investments, capital_instruments`}. Different assets can have attributes, e.g. a claim can have (among other attributes) a `ISSUER` and a `DENOMINATION`. In Appendix C we show an example of such a randomly generated regulation. For our experiments, we can now draw on a large library of randomly generated regulations that participants have to evaluate.

We plan to run the following experiments:

1. Backtesting. Our first step is to check that our measures can explain the performance of the participants to the experiment. So as not to introduce any bias when selecting which instructions to give to the participants, we create a program that generates random instructions to compute risk-weighted assets in a format similar to the Basel I text studied in Section 4 (we manually check that the instructions make sense, e.g., they do not contain contradictory rules), and a bank balance sheet. Each participant is given a different set of instructions and a bank balance sheet and computes the bank's risk-weighted assets. We then regress measures of the participant's performance on different measures of complexity. In particular, we are interested in testing whether measures based on the level have explanatory power over volume. In addition, we can check whether the impact of complexity is different for participants with different backgrounds.

2. Disentangling psychological complexity from computational complexity. To study the difference between these two forms of complexity, we compare three different treatments: (1) The subject is shown a regulation with five different asset classes, and a bank balance sheet containing non-zero values for all five asset classes; (2) The subject is shown the same regulation, but the bank balance sheet contains only three asset classes out of five; (3) The subject is shown a regulation with three asset classes, and the bank balance sheet contains the same three asset classes.

The idea of this experiment is that treatments (1) and (2) should differ in their computational complexity but not in their psychological complexity, conversely (2) and (3) have the same computational complexity but different levels of psychological complexity. Using these two comparisons we can disentangle the role of psychological complexity and computational complexity in explaining the different mistakes made in a regulation with three asset classes relative to a regulation with five asset classes.

3. Disentangling problem complexity from psychological complexity. Here our idea is to compare different regulations solving the same problem. In order to do so we will compare the performance of the participants when the regulation is given in different formats: (1) A high-level format, e.g., "Commercial loans have a risk-weight of X%, which is reduced to Y% if the maturity is less than Y and the counterparty is located in an OECD country." (2) A low-level format, e.g., "Commercial loans with a maturity of more than Y have a risk-weight of X%. Commercial loans with a counterparty not located in an OECD country have a risk-weight of X%. Commercial loans with a maturity less than Y and a counterparty located in an OECD country have a risk-weight of Y%."

The idea of this experiment is that treatments (1) and (2) have the same potential volume V^* and solve the same regulatory problem. Differences in outcomes can thus only come from psychological complexity. Note that since V^* is the same in both cases, the fact that regulation (2) is longer implies that it has a lower level. Observing that subjects perform better on regulation (2) would validate the idea that the level captures a dimension of complexity different from volume and from problem complexity.

4. Shape of the function C. To estimate how a lower level can substitute for a lower volume, we start with a first regulation (1) characterized by a volume V_1 and a level L_1. We then keep the level constant and increase the volume by 25%, giving a second regulation (2) characterized by $(V_2, L_2) = (1.25V_1, L_1)$. We then gradually decrease the level, keeping the volume constant, with regulations (V_3, L_3), (V_4, L_4), etc., until we reach the same outcome (within some confidence bounds). At the end of this process, we have two regulations characterized by (V_1, L_1) and $(1.25V_1, L_n)$ such that $C(V_1, L_1) = C(1.25V_1, L_n)$. The outcome of this process gives us an estimate of the rate of substitution between level and volume. In addition, we can study how this rate of substitution varies across subjects, depending on their studies, professional experience, background, etc.

7 Conclusion

This paper is based on the idea that a financial regulation can be seen as an algorithm that applies a set of instruction to a regulated entity in order to return a regulatory action. The study of regulatory complexity can then be conducted using tools from computer science and aimed at capturing the complexity of algorithms.

The present work is only a first step in applying this new approach to the study of regulatory complexity, and is meant as a "proof of concept". We show how some of the simplest measures of regulatory complexity can be applied to financial regulation, in different contexts: (i) a theoretical model of capital regulation, in which we can compute a theoretically optimal regulation taking into account the cost of its complexity; (ii) an algorithmic "translation" of the Basel I Accords; (iii) the original text of the Basel I Accords; (iv) the original text of the Dodd-Frank Act; (v) experiments using artificial "Basel-I like" regulatory instructions.

While the results we present are very preliminary, we believe they are encouraging and highlight several promising avenues for future research. First, the dictionary that we created will allow other interested researchers to compute various complexity measures for other regulatory texts and compare them to those we produced for Basel I and the Dodd-Frank

Act. Moreover, the dictionary can be enriched in a collaborative way. Such a process would make the measures more robust over time and allow to compare the complexity of different regulatory topics, different updates of the same regulation, different national implementations, etc. This can also serve as a useful benchmarking tool for policymakers drafting new regulations. Second, the conceptual framework and the experiments we propose to separate three dimensions of complexity (problem, psychological, computational) can be used for other measures than ours, and give a framework to conduct "horse races" between different measures of complexity. Finally, our measures could be used in empirical studies aiming at testing what is the impact of regulatory complexity, and in particular testing some of the mechanisms that have been proposed in the theoretical literature.

References

ARORA, S., BARAK, B., BRUNNERMEIER, M. and GE, R. (2009). *Computational Complexity and Information Asymmetry in Financial Products*. Working paper. 4

ASRIYAN, V., FOARTA, D. and VANASCO, V. (2018). *Strategic Complexity*. Working paper. 4

BASEL COMMITTEE ON BANKING SUPERVISION (1988). *International convergence of capital measurement and capital standards*. Discussion paper. 14

BASEL COMMITTEE ON BANKING SUPERVISION (2013). *The regulatory framework: balancing risk sensitivity, simplicity and comparability*. Discussion paper. 1

CALOMIRIS, C. W. (2018). Has financial regulation been a flop? (or how to reform dodd-frank). *Journal of Applied Corporate Finance*, **29** (4), 8–24. 1

CANFORA, G., GARCIA, F., PIATTINI, M., RUIZ, F. and VISAGGIO, C. (2005). A family of experiments to validate metrics for software process models. *Journal of Systems and Software*, **77** (2), 113 – 129. 3

CARLIN, B. I. (2009). Strategic price complexity in retail financial markets. *Journal of Financial Economics*, **91** (3), 278 – 287. 4

CÉLÉRIER, C. and VALLÉE, B. (2017). Catering to investors through security design: Headline rate and complexity. *The Quarterly Journal of Economics*, **132** (3), 1469–1508. 4

ELLISON, S. F. (2016). Price search and obfuscation: an overview of the theory and empirics. In E. Basker (ed.), *Handbook on the Economics of Retailing and Distribution*, Edward Elgar Publishing. 4

HAKENES, H. and SCHNABEL, I. (2012). *Regulatory Capture By Sophistication*. Working paper. 4

HALDANE, A. and MADOUROS, V. (2012). The dog and the frisbee. *Proceedings - Economic Policy Symposium - Jackson Hole*, pp. 109–159. 1, 7

HALSTEAD, M. H. (1977). *Elements of Software Science*. Elsevier. 2, 7, 8, 9

HELLWIG, M. (2010). *Capital Regulation after the Crisis: Business as Usual?* Working paper. 4

LI, W., AZAR, P., LAROCHELLE, D., HILL, P. and LO, A. W. (2015). Law is code: A software engineering approach to analyzing the united states code. *Journal of Business and Technology Law*, **10** (2). 4

MCCABE, T. J. (1976). A complexity measure. *IEEE Transactions on Software Engineering*, **SE-2** (4), 308–320. 5

ROCHET, J.-C. (2010). The future of banking regulation. In M. Dewatripont, J.-C. Rochet and J. Tirole (eds.), *Balancing the Banks*, Princeton University Press, pp. 78–106. 4

RUBINSTEIN, A. (1986). Finite automata play the repeated prisoner's dilemma. *Journal of Economic Theory*, **39** (1), 83 – 96. 4

YU, S. and ZHOU, S. (2010). A survey on metric of software complexity. In *2010 2nd IEEE International Conference on Information Management and Engineering*, pp. 352–356. 5

A Proof of Section 3

Proof of (5). For a given $E \in [a,b]$, total welfare is given by:

$$\begin{aligned} W_{a,b}(E) &= \int_a^b [-\lambda(1-p)\min(x-E,0)-\delta E]dx & (12)\\ &= -\lambda(1-p)\int_E^b (x-E)dx-\delta E(b-a) & (13)\\ &= -\lambda(1-p)\frac{(b-E)^2}{2}-\delta E(b-a). & (14)\end{aligned}$$

Maximizing this quantity with respect to E gives the desired result.

Proof that intervals optimally have the same length. Consider the case of two intervals, $[0,\bar{x}]$ and $[\bar{x},1]$. Total expected welfare is then given by:

$$\begin{aligned} W_{0,\bar{x}}(E^*_{0,\bar{x}})+W_{\bar{x},1}(E^*_{\bar{x},1}) &= \delta\bar{x}\left[\frac{\delta\bar{x}}{2\lambda(1-p)}-\bar{x}\right]+\delta(1-\bar{x})\left[\frac{\delta(1-\bar{x})}{2\lambda(1-p)}-1\right] & (15)\\ &= \delta\bar{x}(1-\bar{x})\frac{\lambda(1-p)-\delta}{\lambda(1-p)}-\frac{\delta}{2\lambda(1-p)}[\delta-2\lambda(1-p)]. & (16)\end{aligned}$$

We immediately see that the optimal $\bar{x}$ is equal to $1/2$, that is, the two intervals are symmetric.

Consider now any number I of intervals. Following the same approach it is easily proved that all intervals must have the same length, so that the I intervals are $[0,1/I]$, $[1/I,2/I]\ldots[(I-1)/I,1]$. The $i+1$-th interval has a welfare of:

$$\begin{aligned} W_{i/I,(i+1)/I}(E^*_{i/I,(i+1)/I}) &= \frac{\delta}{I}\left[\frac{\delta}{2I\lambda(1-p)}-\frac{i+1}{I}\right] & (17)\\ &= \frac{\delta}{I^2}\left[\frac{\delta-2\lambda(1-p)}{2\lambda(1-p)}-i\right]. & (18)\end{aligned}$$

We use this expression to compute (7).

B Basel I Algorithm

In the following, we provide a description of the Basel I regulation in the form of a stylized algorithm. We use pseudo code that simply captures the logical flow of the instructions in Basel I.

```
IF (
    ASSET_CLASS == ''cash'' OR
    ASSET_CLASS == ''claims'' AND (
        (ISSUER == ''central governments'' OR ISSUER == ''central banks'') AND
        DENOMINATION == ''national'' AND
        FUNDING_CURRENCY == ''national''
    ) OR
    ASSET_CLASS == ''claims'' AND (
        (ISSUER == ''central governments'' OR ISSUER == ''central banks'') AND
        ISSUER_COUNTRY == ''oecd''
    ) OR
    ASSET_CLASS == ''claims'' AND (
        (COLLATERALIZED == ''oecd'' OR GUARANTEED == ''oecd'')
    )
) THEN:
    risk_weight = 0.0;

ELSE IF (
    ASSET_CLASS == ''claims'' AND (
        (ISSUER == ''public-sector entities'' AND ISSUER_COUNTRY == ''domestic'') AND
        (ISSUER != ''central governments'' AND ISSUER_COUNTRY == ''domestic'')
    ) OR
    ASSET_CLASS == ''loans'' AND (
        (GUARANTEED == ''public-sector entities'' AND GUARANTEED_COUNTRY == ''domestic'') AND
        (GUARANTEED != ''central governments'' AND GUARANTEED_COUNTRY == ''domestic'')
    )
) THEN:
    risk_weight = national_discretion;

ELSE IF (
    ASSET_CLASS == ''claims'' AND (
        (ISSUER == ''IBRD'' OR ISSUER == ''IADB'' OR ISSUER == ''AsDB'' OR ISSUER == ''AfDB'' OR
            ISSUER == ''EIB'') OR
        (GUARANTEED == ''IBRD'' OR GUARANTEED == ''IADB''  OR GUARANTEED == ''AsDB'' OR
            GUARANTEED == ''AfDB'' OR GUARANTEED == ''EIB'') OR
```

```
        (COLLATERALIZED == ''IBRD'' OR COLLATERALIZED == ''IADB''  OR COLLATERALIZED == ''AsDB'' OR
            COLLATERALIZED == ''AfDB'' OR COLLATERALIZED == ''EIB'')
    ) OR
    ASSET_CLASS == ''claims'' AND ( // 3b
        (ISSUER == ''bank'' AND ISSUER_COUNTRY == ''oecd'')
    ) OR
    ASSET_CLASS == ''loans'' AND (
        (GUARANTEED == ''bank'' AND GUARANTEED_COUNTRY == ''oecd'')
    ) OR
    ASSET_CLASS == ''claims'' AND (  // 3c
        (ISSUER == ''bank'' AND ISSUER_COUNTRY != ''oecd'' AND ASSET_MATURITY <= 1)
    ) OR
    ASSET_CLASS == ''loans'' AND (
        (GUARANTEED == ''bank'' AND GUARANTEED_COUNTRY != ''oecd'' AND ASSET_MATURITY <= 1)
    ) OR
    ASSET_CLASS == ''claims'' AND ( // 3d
        (ISSUER == ''public sector entities'' AND ISSUER != ''central governments'' AND
            ISSUER_COUNTRY == ''oecd'' AND ISSUER_COUNTRY != ''domestic'')
    ) OR
    ASSET_CLASS == ''loans'' AND (
        (GUARANTEED == ''public sector entities'' AND GUARANTEED != ''central governments'' AND
            GUARANTEED_COUNTRY == ''oecd'' AND GUARANTEED_COUNTRY != ''domestic'')
    ) OR
    ASSET_CLASS == ''cash'' AND ( // 3e
        CASH_COLLECTION == ''in process''
    )
) THEN:
    risk_weight = 0.2;

ELSE IF (
    ASSET_CLASS == ''loans'' AND
        (LOAN_SECURITY == ''mortgage'' AND (PROPERTY_OCCUPIED == ''owner'' OR
            PROPERTY_OCCUPIED == ''rented''))
) THEN:
    risk_weight = 0.5;
```

```
ELSE IF (
    ASSET_CLASS == ''claims'' AND (
        ISSUER == ''private sector''
    ) OR
    ASSET_CLASS == ''claims'' AND (
        (ISSUER == ''banks'' AND ISSUER_COUNTRY != ''oecd'' AND ASSET_MATURITY > 1)
    ) OR
    ASSET_CLASS == ''claims'' AND (
        (ISSUER == ''central governments'' AND ISSUER_COUNTRY != ''oecd'' AND
            DENOMINATION != ''national'' AND FUNDING_CURRENCY != ''national'')
    ) OR
    ASSET_CLASS == ''claims'' AND (
        (ISSUER == ''commercial companies'' AND ISSUER_OWNER == ''public sector'')
    ) OR
    (ASSET_CLASS == ''premises'' OR ASSET_CLASS == ''plant'' OR ASSET_CLASS == ''equipment'' OR
        ASSET_CLASS == ''other fixed assets'') OR
    (ASSET_CLASS == ''real estate'' OR ASSET_CLASS == ''other investments'') OR
    ASSET_CLASS == ''capital instruments'' AND (
        (ISSUER == ''banks'' AND DEDUCTED_FROM != ''capital'')
    )
) THEN:
  risk_weight = 1.0;

ELSE:
    risk_weight = 1.0;
```

C Randomly Generated Regulations

```
IF (
    ASSET_CLASS == ''real_estate''  OR
    ASSET_CLASS == ''other_investments''
) THEN:
    risk_weight = 0.0;
ELSE IF (
```

```
    ASSET_CLASS == ''other_investments''  OR
    ASSET_CLASS == ''real_estate''  OR
    ASSET_CLASS == ''capital_instruments'' AND
       (ISSUER == ''central governments'' AND ISSUER == ''AsDB'') OR
    ASSET_CLASS == ''real_estate''  OR
    ASSET_CLASS == ''plant''
) THEN:
    risk_weight = 0.2;
ELSE IF (
    ASSET_CLASS == ''other_fixed_assets''
) THEN:
    risk_weight = 0.2;
ELSE IF (
    ASSET_CLASS == ''plant''  OR
    ASSET_CLASS == ''real_estate''  OR
    ASSET_CLASS == ''other_fixed_assets''  OR
    ASSET_CLASS == ''plant''  OR
    ASSET_CLASS == ''claim'' OR
       (COLLATERALIZED == ''IADB'' AND COLLATERALIZED == ''oecd'' OR COLLATERALIZED == ''AsDB''
        AND COLLATERALIZED == ''IADB'' AND COLLATERALIZED == ''AsDB''
        OR COLLATERALIZED == ''EIB'')
) THEN:
    risk_weight = 0.2;
ELSE IF (
    ASSET_CLASS == ''cash'' OR
    ASSET_CLASS == ''real_estate''  OR
    ASSET_CLASS == ''premises''  OR
    ASSET_CLASS == ''equipment''  OR
    ASSET_CLASS == ''equipment''  OR
    ASSET_CLASS == ''plant''
) THEN:
    risk_weight = 0.2;
ELSE:
    risk_weight = 1.0;
```