DISCUSSION PAPER SERIES

DP14377

MEASURING REGULATORY COMPLEXITY

Jean-Edouard Colliard and Co-Pierre Georg

FINANCIAL ECONOMICS

ISSN 0265-8003

MEASURING REGULATORY COMPLEXITY

Jean-Edouard Colliard and Co-Pierre Georg

Discussion Paper DP14377
Published 02 February 2020
Submitted 30 January 2020

Centre for Economic Policy Research
33 Great Sutton Street, London EC1V 0DX, UK
Tel: +44 (0)20 7183 8801
www.cepr.org

MEASURING REGULATORY COMPLEXITY

Abstract

Despite a heated debate on the perceived increasing complexity of fi nancial 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 like 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 to the regulation of a bank in a theoretical model, to an algorithm computing capital requirements based on Basel I, and to actual regulatory texts. Our measures capture dimensions of complexity beyond the mere length of a regulation. In particular, shorter regulations are not necessarily less complex, as they can also use more "high-level" language and concepts. Finally, we propose an experimental protocol to validate measures of regulatory complexity.

JEL Classification: G18

Keywords: financial regulation, Capital regulation, Regulatory Complexity, Basel Accords

Jean-Edouard Colliard - colliard@hec.fr
HEC Paris and CEPR

Co-Pierre Georg - co-pierre.georg@bundesbank.de
University of Cape Town & Deutsche Bundesbank

Acknowledgements
We are grateful to Igor Kozhanov and Iman van Lelyveld, as well as to participants to the 2019 BCBS-CEPR Workshop on "Impact of regulation in a changing world: innovations and new risks", the 2019 NY Fed Fintech Conference, the 2016 "Journée de la Chaire ACPR", seminar audiences at Autorité des Marchés Financiers, HEC, the University of Bonn, the University of Exeter, 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.

Measuring Regulatory Complexity *

Jean-Edouard Colliard†
HEC Paris

Co-Pierre Georg‡
University of Cape Town
Deutsche Bundesbank

January 20, 2020

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 like 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 to the regulation of a bank in a theoretical model, to an algorithm computing capital requirements based on Basel I, and to actual regulatory texts. Our measures capture dimensions of complexity beyond the mere length of a regulation. In particular, shorter regulations are not necessarily less complex, as they can also use more "high-level" language and concepts. Finally, we propose an experimental protocol to validate measures of regulatory complexity.

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

JEL classification: G18, G28, G41.

*We are grateful to Igor Kozhanov and Iman van Lelyveld, as well as to participants to the 2019 BCBS-CEPR Workshop on "Impact of regulation in a changing world: innovations and new risks", the 2019 NY Fed Fintech Conference, the 2016 "Journée de la Chaire ACPR", seminar audiences at Autorité des Marchés Financiers, HEC, the University of Bonn, the University of Exeter, 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).[1]

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 informative, 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 observe that a regulation can essentially be seen as an algorithm performing some operation: it is a list of instructions that are applied to an economic agent and returns a regulatory action (e.g., a sanction). In other words, we propose to interpret any financial regulation as an algorithm that takes financial institutions as inputs and returns a regulatory action as an output. This parallel opens the possibility of using an extensive literature in computer science on algorithmic complexity and apply it to the study of regulatory complexity. The complexity of a regulation is then measured by the complexity of the associated algorithm.

As a proof of concept, in this paper we start with simple measures proposed in the

[1] Gai *et al.* (2019) provide a comprehensive discussion of the policy issues at stake.

computer science literature. We apply these measures in a variety of contexts. We first use them to compute the complexity of an artificial regulation in a normative model of bank regulation. We then "translate" an actual regulation–the Basel I capital requirements–into a functioning algorithm, and compute measures of the complexity of this algorithm. We also compute these measures based on the regulatory text itself, both for the Basel I capital requirements and for the Dodd-Frank Act, a much broader text. Finally, we propose a novel experimental protocol to test the relevance of our measures of regulatory complexity, or any other measure.[2]

The first section of the paper also introduces a framework to formally define what is a measure of regulatory complexity, and the different dimensions of complexity that can be captured. In particular, we make a distinction between: (i) "problem complexity", a regulation is complex because it aims at imposing many different rules on the regulated entities; (ii) "psychological complexity", a regulation is complex because it is difficult for a human reader to understand; (iii) "computational complexity", a regulation is complex because it is long and costly to implement. Given that our measures rely on the analysis of the text describing a regulation, they can capture problem complexity and psychological complexity, but not computational complexity.

We then introduce the new measures we propose. Among the many measures of algorithmic complexity that have been studied in the computer science literature, we focus in this paper on the measures pioneered by Halstead (1977). As we detail in Section 2, these "Halstead 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 two factors. First, these measures

[2]The experiments will ultimately be part of this paper.

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. Second, 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.

We then apply these measures in different contexts. To motivate why properly measuring regulatory complexity is important, we first study a simple theoretical model of capital regulation in which there is a trade-off between the complexity and the risk-sensitivity of regulation, as claimed by Basel Committee on Banking Supervision (2013). In our model, 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. We obtain in particular that the complexity of the regulation is concave in the number of risk buckets: our measures capture the fact that a regulation based on this structure is intrinsically repetitive, and that each new risk bucket is less complex at the margin. More generally, 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 show how to measure the complexity of such a capital regulation in practice by considering the design of risk weights in the Basel I Accords. 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 a computer code corresponding to the instructions of Basel I and measure the algorithmic complexity of this code, that is, we use the measures of algorithmic complexity literally; (ii) We analyze the 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 an actual 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, and given the encouraging results obtained with the Basel I Accords, we show how the measures can be computed at a much larger scale by applying our text analysis approach to the different titles of the 2010 Dodd-Frank Act. We 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 very 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.

Fourth, and finally, we describe an experimental protocol that can be used to test the power of our measures. Experimental subjects are given a regulation consisting in (randomly generated) Basel-I type rules, and the balance sheet of a bank. They have to compute the capital ratio of a bank and to say whether the bank satisfies the regulatory threshold. The power of a measure of regulatory complexity is given by its ability to forecast whether a subject returns a wrong value of the capital ratio, and the time taken to answer. Moreover, we can test whether the relation between the measure of regulatory complexity and the outcome depends on the student's background and training, etc. In this preliminary version of the paper, we only outline an experimental protocol and leave the conduct of the experiment for future research. Importantly, our protocol can be used to validate any measure of regulatory complexity based on the text of a regulation, not only ours, and thus opens the path to comparing the performance of different measures.[3].

This paper is part of a growing literature using text analysis to quantify the effects of regulatory change. Al-Ubaydli and McLaughlin (2017) focus on the wording of regulations

[3]Ultimately the objective would be to establish a standard method to measure the power of a measure of complexity. This has been done in computer science, in which there is a literature testing whether different measures of algorithmic complexity correlate with mistakes made by the programmers or the time they need to code the program (see, e.g., Canfora *et al.* (2005))

and quantify binding constraints in the Code of Federal Regulations for all industries and regulatory agencies in the United States. Focusing on financial regulation, Barth and Miller (2018) study the capital adequacy guidelines following the Latin American Debt Crisis of 1982 and the Global Financial Crisis 2007/2008 and show that these guidelines have become more complex over time. They follow the methodology of Herring (2018), who measures complexity through the number of mathematical operations involved in the computation of capital adequacy ratios for Global Systemically Important Banks (G-SIBs). This approach is in line with the original idea of Haldane and Madouros (2012) to measure the complexity of regulation through its length. Our paper differs from this approach in two important ways. First, we distinguish between problem- and psychological complexity, i.e. between the complexity inherent in the problem itself and the complexity arising from the specific formulation of the problem in natural language. And second–and related–we can distinguish the complexity of texts of similar length but with different instructions.

The documented increase in the "complexity" (or perhaps more appropriately, the extent) of financial regulation has real consequences, as a growing literature shows. Kalmenovitz (2019) constructs a daily measure of regulatory intensity and shows that increased regulatory intensity leads to a significant reduction in firm-level investment and hiring. Gutiérrez and Philippon (2019) argue that the increase in regulation can account for the decline in the elasticity of entry with respect to Tobin's Q since the late 1990s.

We contribute to this literature in three ways. First, we introduce a new measure that creates a bridge between the question of regulatory complexity and the extent literature on regulatory complexity. Second, we propose a theoretical framework that allows to define different dimensions of regulatory complexity and classify different measures according to the dimension they capture. Third, we propose an experimental protocol that can be used to test different measures of complexity.

There is also a growing empirical literature proposing different measures for the complexity of various economic objects, such as financial products (see 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)).[4] 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.

A number of theory papers have studied 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. We hope that progress in the measurement of regulatory complexity may lead to empirical tests of these theories. 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. 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. To our knowledge, none of these theories has been tested yet, in particular due to the lack of measures of regulatory complexity.

Our methodology is related to a literature in behavioral economics that models economics agents as computer programs, in particular Rubinstein (1986), who 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

[4] Amadxarif *et al.* (2019) use a hybrid approach between the network analysis of Li *et al.* (2015) and natural language processing. Their definition of complexity is based on the difficulty of processing "linguisitc units".

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.

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

The term "complexity" being somewhat vague, different authors, policymakers and industry participants have different concepts in mind when referring to "regulatory complexity". In this section we set preliminary definitions so as to clarify some of the different dimensions of complexity, and 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". Knuth (1973) describes an algorithm as *"a finite set of rules that gives a sequence of operations for solving a specific type of problem"* and identifies five features an algorithm must satisfy. First, an algorithm must terminate after a finite number of steps. Second, each step of the algorithm must be precisely defined—be it verbally or through formal use of programming languages. Third, an algorithm has zero or more inputs, taken from a well specified set of objects. Fourth, it has one or more outputs, quantities that have a specified relationship to the inputs. Lastly, an algorithm should use sufficiently simple operations so that it can be computed, in principle, *"by someone using pencil and paper."* Surprisingly, a formal definition of an algorithm beyond the informal characterization provided above is not without difficulty.[5] For the purpose of our paper, the informal description of an algorithm provided by Knuth (1973) is sufficient.

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 (e.g., letting the bank operate, imposing a fine...). Formally, we define:

Definition 1. *A regulatory problem is a mapping $f : \mathcal{E} \to \Sigma$ from the set of regulated entities $\mathcal{E}$ 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 $\mathcal{V}$. 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 and/or for the regulated entity. We call this last step "supervision":

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

We can now give defining properties of measures of regulatory complexity corresponding to different dimensions. Assume that we have a set $\tilde{\mathcal{F}} = \{\tilde{f}_1, \tilde{f}_2..., \tilde{f}_n\}$ of regulations

[5] The first feature of an algorithm, the requirement that it terminates, is where problems arise. Otherwise, an algorithm could in principle be defined using a Turing machine (see, for example, the definition in (Arora and Barak, 2009)).

solving the same regulatory problem f, and a set $\mathcal{E} = \{e_1, e_2, ..., e_m\}$ 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 $\mu : \tilde{\mathcal{F}} \times \mathcal{E} \to \mathbb{R}$.*

If μ is a measure of problem complexity, then $\mu(\tilde{f}, e)$ is constant in $\tilde{f}$ and e.

If μ is a measure of psychological complexity, then $\mu(\tilde{f}, e)$ is constant in e but not necessarily in $\tilde{f}$.

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 vector norm of an n-dimensional vector $x = (x_1, x_2 ... x_n)$ can be written as:

```
y = sqrt(x_1^2+x_2^2...+x_n^2)
```

Here, the operators are $=, sqrt, +, \hat{}$, and the operands $y, x_i, 2$. So we have $\eta_1 = 4$, $N_1 = 2n+1$, $\eta_2 = n+2$, $N_2 = 2n+2$.

A simple measure of complexity, corresponding to the length, is simply the total number of operators and operands, called the *volume*:[6]

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 to this question is to look at the shortest possible 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 vector norm, the shortest possible program is:

```
y = vecnorm(x_1,x_2...x_n)
```

where "vecnorm" is a function returning the vector norm. This is the shortest program because any program to compute the norm of a vector would need to specify the input, the output, an assignment rule, and an operation (here, an 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 $2 + \eta_2^*$.*

[6] 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.

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 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:

Assumption 1. *Psychological complexity is captured by a function $C(V, L)$, the cost of complexity, 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.

We assume that the social cost of capital is lower than the expected gain of reducing losses to the public sector:

$$\lambda(1 - p) > \delta. \tag{2}$$

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

$$pr(x) + (1 - p)[1 - x - \lambda \min(x - E, 0)] - \delta E. \tag{3}$$

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:

$$\mathcal{W}(E,x) = -\lambda(1-p)\min(x-E,0) - \delta E. \tag{4}$$

As long as $E < x$, we have $\partial W/\partial E = \lambda(1-p) - \delta$, which by assumption is positive. 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:

$$\int_0^1 \mathcal{W}(x,x)dx = \int_0^1 -\delta x dx = -\frac{\delta}{2}. \tag{5}$$

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 B, for a given interval $[a, b]$ the optimal capital requirement $E^*_{a,b}$ is given by:

$$E^*_{a,b} = b - \delta\frac{b-a}{\lambda(1-p)}. \tag{6}$$

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:

$$W_{a,b}(E^*_{a,b}) = \delta(b-a)\left[\frac{\delta(b-a)}{2\lambda(1-p)} - b\right]. \tag{7}$$

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:

$$\mathcal{W}(I) = \sum_{i=0}^{I-1} W_{i/I,(i+1)/I}(E^*_{i/I,(i+1)/I}) = -\frac{\delta}{2} - \frac{\delta}{2I\lambda(1-p)}[\lambda(1-p) - \delta]. \tag{8}$$

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), "$\geq$" ($2I$ times), "$\leq$" ($I-1$ 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:

$$L(I) = \frac{V^*}{V} = \frac{2I+4}{11I-3}. \tag{9}$$

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

$$L = \frac{2+\eta_2}{\eta_1+\eta_2} \times \frac{\eta_1+\eta_2}{N_1+N_2} = \frac{2I+4}{2I+6} \times \frac{2I+6}{11I-3}. \tag{10}$$

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.

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:

$$I^* = arg \max_I \mathcal{W}(I) - C(V(I), L(I)). \tag{11}$$

The first-order condition can be expressed as:

$$\frac{\delta[\lambda(1-p)-\delta]}{2\lambda(1-p)} \times \frac{1}{I^{*2}} - V'(I^*)C_1(V(I^*), L(I^*)) - L'(I^*)C_2(V(I^*), L(I^*)) = 0. \tag{12}$$

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. Thus, there is a rationale for choosing simpler regulations, even if they do not maximize the economic welfare $\mathcal{W}(I)$. 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 regulations that look simpler (lower I) are actually associated with a higher level of complexity, because L is decreasing in I. This illustrates that it is important to measure different dimensions of complexity and to try to estimate the shape of the function C. Otherwise, efforts to make regulations shorter may end up being counter-productive if this is done by resorting to more high-level language.

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 assets 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 a "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 C. 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;
```

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

Item	V	V^*	L	Contribution to level (in %)
1a	1	3	3.00	-0.34
1b	17	9	0.53	-6.03
1c	13	8	0.62	-4.55
1d	9	6	0.67	-3.10
2a	35	11	0.31	-13.26
3a	61	11	0.18	-11.97
3b	19	10	0.53	-6.79
3c	27	12	0.44	-9.93
3d	35	12	0.34	-13.26
3e	5	5	1.00	2.72
4a	13	8	0.62	6.82
5a	5	5	1.00	0.51
5b	13	9	0.69	-4.55
5c	17	10	0.59	-6.03
5d	9	7	0.78	3.62
5e	7	6	0.86	6.51
5f	3	4	1.33	3.38
5g	9	7	0.78	3.62
5h	1	2	2.00	-0.34
Total	299	46	0.15	0.00

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`, `==`, `>`, $\leq$, and `!=`. Given our algorithmic representation of Basel I, we find that $N_1 = 172, N_2 = 184, \eta_1 = 8, \eta_2 = 45$.

To give a sense to these numbers, we can go further by computing how much the regulation of different asset classes contribute 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.[7] 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 heterogeneous in terms of their contributions to the total level. If we look at the extreme cases, excluding item

[7]Note that each item is not by itself a proper algorithm. For instance, item 1a is simply "cash". As a result, the level can be higher than one.

3d for instance would increase total level by 13.6%, whereas excluding item 4a would decrease the total level by 6.82%. Finally, the level of the entire regulation is 0.15, much smaller than the level of almost all individual rules. This captures the repetitive nature of the Basel I regulation, which is a list of rules structured in the same way and using the same operands multiple times.

4.2 Text analysis

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 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.

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.

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

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

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	249	76	0.30	0.00

the two cases. Table 4 gives the correlation coefficients for the different measures.

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

	Correlation
V	0.64
V^*	0.84
L	0.80
Contribution to level	0.76

The correlation coefficients we obtain are quite large, which shows that the text-based analysis and the algorithm-based analysis are capturing similar patterns. 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 more explicit). The volume is less correlated, and as a result the level (equal to V/V^*) shows a slightly lower but still high correlation.

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 potential volume and 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 prove that our measures can be implemented on a larger scale, 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.[8]

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

[8]We have made the code for for this project, and specifically the dashboard we developed to ease the classification task undertaken, available at `https://github.com/cogeorg/RegulatoryComplexity`. Figure 2 in Appendix A shows the dashboard used to classify individual words. The full list of word classifications can be found here.

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 Section 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 1 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.

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

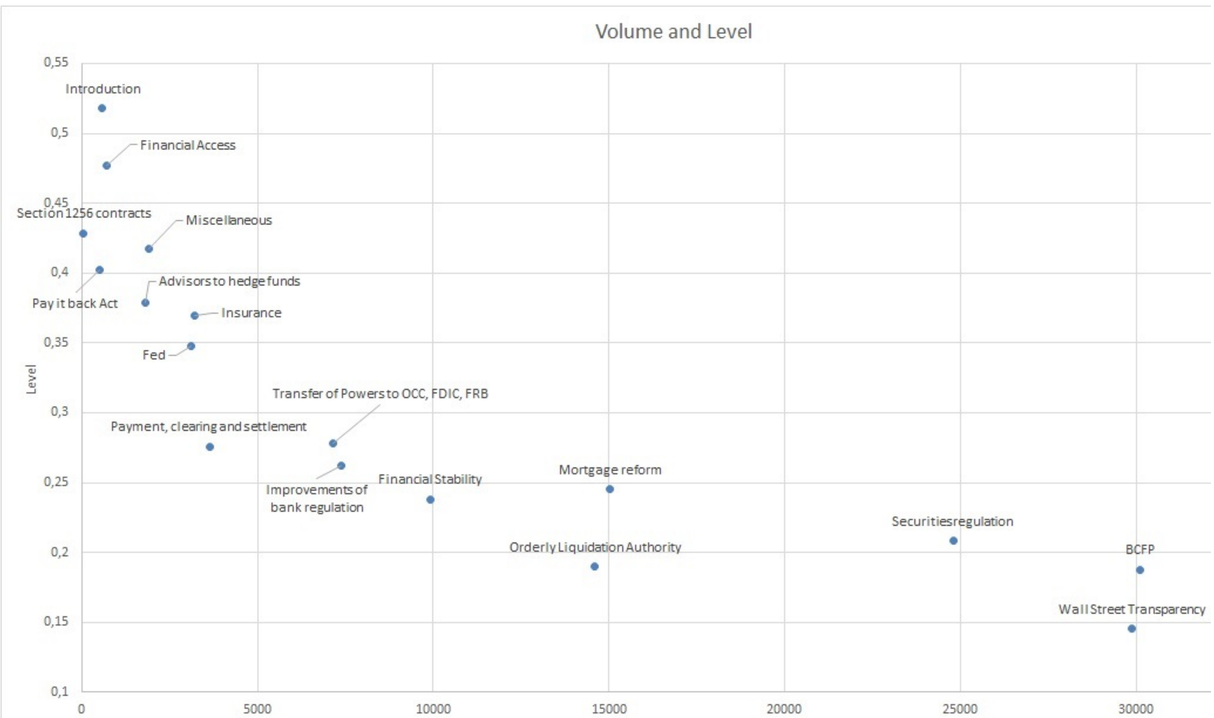


Figure 1: Volume and Level of the different sections of the Dodd-Frank Act.

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

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. Moreover, our protocol can be used for any other text-based measure of complexity and thus would allow in the future to run horse races between different measures.

Second, since it is clear that regulatory complexity is a multidimensional object, we want to shed some light on which dimensions seem to matter more experimentally. Do subjects react more to psychological complexity, problem complexity, or computational complexity? We can also study the heterogeneity of the subjects' reaction to complexity, and how it is determined for instance by background, education, professional experience, etc.

We first describe the general experimental protocol that we propose to follow. We then discuss the detailed tests that we intend to conduct in order to: (i) test that our measures have explanatory power; (ii) test whether computational complexity explains performance in addition to psychological complexity; (iii) test whether psychological complexity explains performance in addition to problem complexity.

6.1 Experimental protocol

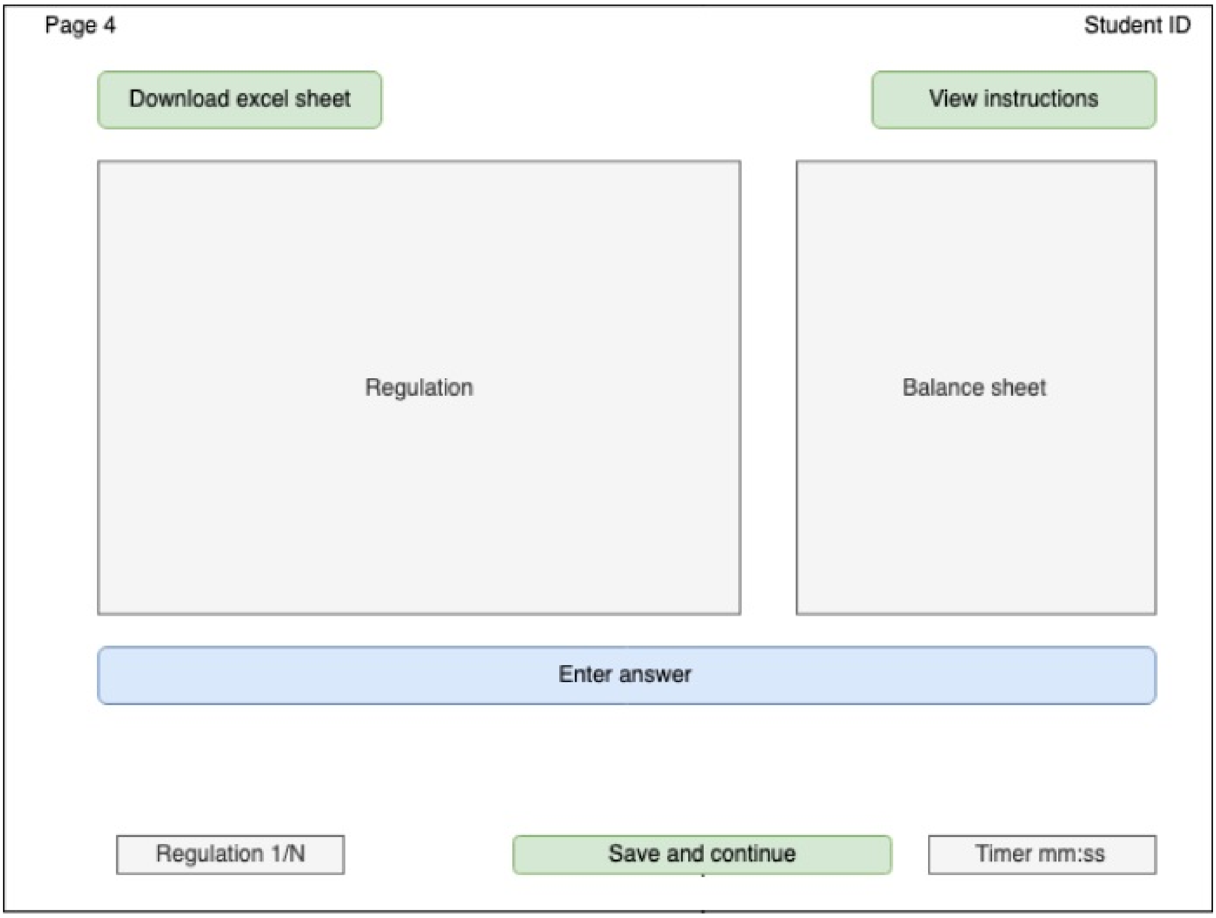
Our experiment is based on the analysis of Basel-I like regulations, as studied in Sections 3 and 4.1, and runs as follows.

We take N participants (business school and engineering students), with N_P being in the range 100-200. Each participant is shown a computer screen which is split vertically in two (see Fig.??). On the left-hand side, there is a series of instructions that mimick a Basel-I like capital regulation (a set of instructions involving different asset classes, characteristics, risk weights, etc.). On the right-hand side, there is a simplified bank balance sheet with details about the different assets of the bank that correspond to the regulation. The participant has to compute the capital ratio of the bank following the instructions. Each participant also has access to an Excel spreadsheet (given empty to the participant), and to a slide containing explanations of the exercise. We record the answer given by the participant (and hence whether it is correct), as well as the time taken to answer.

The participant then moves on to a second regulation and repeats the process, until N_R regulations have been considered, with N_R being around 10. At the end of this process the participant receives a monetary compensation that increases linearly with the number of correct answers given. The exact scheme will be calibrated based on some preliminary trials, the aim being that participants receive a sure payoff of 10 EUR per hour on average, plus a bonus of 20 EUR for the average performance.[9]

At the end of this process we thus have $N \times N_R$ observations, with the following variables: (i) identity of the participant (and hence basic demographic characteristics, educational background, etc.); (ii) regulation considered, in particular its complexity measures; (iii) answer given by the participant, in particular whether it is correct; (iv) time taken to give the answer. As we detail below, we can use these data to compute how much of

[9]Students are typically paid between 10 and 20 euros in such experiments. Given the difficulty of the task we feel it is necessary to slightly increase the remuneration in order to make the experiment attractive.



Page 4
Student ID
Download excel sheet
View instructions
Regulation
Balance sheet
Enter answer
Regulation 1/N
Save and continue
Timer mm:ss

the variation in the accuracy and speed of the answers can be captured by our measures, which offers a test of their validity.

One last important methodological issue is how to choose the regulatory texts given in the experiments. We propose to generate a database of random variations "around" the original Basel I rules, e.g., by deleting some instructions, simplifying the extra conditions associated with different asset classes, etc. More precisely, 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`. As a last step, we manually check that the instructions make sense, e.g., they do not contain contradictory rules. In Appendix D we show an example of such a randomly generated regulation.

We thus obtain large library of randomly generated regulations, that all have different complexity measures, and that participants can use in the experiment. Importantly, because these regulations are generated randomly, we have no control over their exact wording. In particular, this alleviates any potential concern that we may handpick regulatory instructions so as to get results supporting the validity of our measures. We will then also generate random bank balance sheets in which the different assets correspond to those identified in the regulatory instructions.

6.2 Evaluating complexity measures

Our first step is to check that our measures can explain the performance of the participants to the experiment. After N_R iterations of the experiment with each of the N participants, we obtain $N \times N_R$ observations at the participant $\times$ iteration level. We can evaluate the power of our complexity measures by running the following regression:

$$x_{i,t} = \alpha + \beta V_{i,t} + \gamma L_{i,t} + \eta_i + \nu_t + \epsilon_{i,t}, \tag{13}$$

where $x_{i,t}$ is a measure of the performance of participant i at iteration number t, for instance a dummy variable equal to 1 if the answer is correct, or 1 over the time taken to provide a correct answer; $V_{i,t}$ and $L_{i,t}$ are the volume and level of the regulation considered in that iteration; η_i is a participant fixed effect capturing unobserved heterogeneity among participants, and ν_t is an iteration fixed effect, capturing for instance the possibility that participants may obtain better results over time (learning), or on the contrary worse results (tiredness).

The predictive power of our complexity measures can be evaluated by comparing the R^2 of regression (13) to the R^2 of a regression without $V_{i,t}$ and $L_{i,t}$, or without $L_{i,t}$ but with $V_{i,t}$. In addition, we can test that the signs are correct, that is, β and γ should both be negative.

Note that the specification (13) can be interpreted as a particular complexity cost function C. We can test variants of this specification to try to learn more about the shape of that function, for instance by introducing non-linear terms. A particularly interesting exercise would be to introduce interaction terms between V or L and participants' characteristics. For instance, it is possible that students with a business background and an engineering background react differently to the level and to the volume.

6.3 Disentangling psychological complexity from computational complexity

In our experiments, participants face at the same time problem complexity (the task they have to perform), psychological complexity (understanding the instructions), and computational complexity (performing the operations necessary to provide the correct answer). An advantage of an experimental setup is that it can be used to generate clean tests to disentangle these different dimensions.

In particular, this is how we propose to disentangle psychological complexity from computational complexity. We start with $3N$ participants, and allocate them to three groups of size N each. We will iterate N_R times. In the first iteration, subjects in group 1 are shown a regulation with five different asset classes, and a bank balance sheet containing non-zero values for all five asset classes (treatment A). Subjects in group 2 are shown the same regulation, but the bank balance sheet contains only three asset classes out of five (treatment B). Subjects in group 3 are shown a regulation with three asset classes, and the bank balance sheet contains the same three asset classes (treatment C). We then move to the second iteration, in which we repeat the same experiment with a new regulation, with group 1 now receiving treatment B, group 2 treatment C, and group 3 treatment A. We rotate again at each iteration.

Under the null hypothesis that computational complexity plays no role, there should be no statistically significant difference in performance between treatments A and B. Indeed, they differ in their computational complexity but not in their psychological complexity or their problem complexity. Under the null hypothesis that psychological complexity plays no role, there should be no difference in performance between treatments B and C, as they have the same computational complexity and problem complexity but not the same psychological complexity.

Here again one could additionally explore the cross-sectional heterogeneity of the participants, as possibly different dimensions of complexity matter more for different profiles of participants.

6.4 Disentangling psychological complexity from problem complexity

We follow a similar approach to separate psychological complexity and problem complexity. Here our idea is to compare different regulations solving the same problem. We start with $2N$ participants and allocate them to two groups of size N. We interate N_R times. In the first iteration, subjects in group 1 are shown a regulation in 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." Subjects in group 2 are shown an equivalent regulation in 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 also validate the idea that the level captures a dimension of complexity different from volume and from problem complexity.

7 Conclusion

This paper is based on the idea that a financial regulation can be seen as an algorithm that applies a set of instructions 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 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 applied to other measures that have been proposed in the literature, so as to better understand what each one is capturing. 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

Al-Ubaydli, O. and McLaughlin, P. A. (2017). Regdata: A numerical database on industry-specific regulations for all united states industries and federal regulations, 1997–2012. *Regulation and Governance*, **11**, 109–123. 4

Amadxarif, Z., Brookes, J., Garbarino, N., Patel, R. and Walczak, E. (2019). *The language of rules: textual complexity in banking reforms*. Staff Working Paper No.834, Bank of England. 6

Arora, S. and Barak, B. (2009). *Computational Complexity: A Modern Approach*. New York, NY, USA: Cambridge University Press, 1st edn. 8

—, —, Brunnermeier, M. and Ge, R. (2009). *Computational Complexity and Information Asymmetry in Financial Products*. Working paper. 6

Asriyan, V., Foarta, D. and Vanasco, V. (2018). *Strategic Complexity*. Working paper. 6

Barth, J. R. and Miller, S. M. (2018). On the rising complexity of bank regulatory capital requirements: From global guidelines to their united states (us) implementation. *Journal of Risk and Financial Management*, **11** (4). 5

Basel Committee on Banking Supervision (1988). *International convergence of capital measurement and capital standards*. Discussion paper. 17

Basel Committee on Banking Supervision (2013). *The regulatory framework: balancing risk sensitivity, simplicity and comparability*. Discussion paper. 1, 3

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. 4

Carlin, B. I. (2009). Strategic price complexity in retail financial markets. *Journal of Financial Economics*, **91** (3), 278 – 287. 6

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. 5

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. 5

Gai, P., Kemp, M., Sanchez Serrano, A. and Schnabel, I. (2019). *Regulatory complexity and the quest for robust regulation*. Report of the Advisory Scientific Committee 8, European Systemic Risk Board. 1

Gutiérrez, G. and Philippon, T. (2019). *The Failure of Free Entry*. Working paper w26001, NBER. 5

Hakenes, H. and Schnabel, I. (2012). *Regulatory Capture By Sophistication*. Working paper. 6

Haldane, A. and Madouros, V. (2012). The dog and the frisbee. *Proceedings - Economic Policy Symposium - Jackson Hole*, pp. 109–159. 1, 5, 10

Halstead, M. H. (1977). *Elements of Software Science*. Elsevier. 2, 10, 11, 12

Hellwig, M. (2010). *Capital Regulation after the Crisis: Business as Usual?* Working paper. 6

Herring, R. J. (2018). The evolving complexity of capital regulation. *Journal of Financial Services Research*, **53** (2), 183–205. 5

Kalmenovitz, J. (2019). *Regulatory Intensity*. mimeo, NYU Stern School of Business. 5

Knuth, D. E. (1973). *Fundamental Algorithms: The Art of Computer Programming*, vol. 1. Addison-Wesley, 2nd edn. 7, 8

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). 6

McCabe, T. J. (1976). A complexity measure. *IEEE Transactions on Software Engineering*, **SE-2** (4), 308–320. 7

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. 6

Rubinstein, A. (1986). Finite automata play the repeated prisoner's dilemma. *Journal of Economic Theory*, **39** (1), 83 – 96. 6

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. 7

A Figures

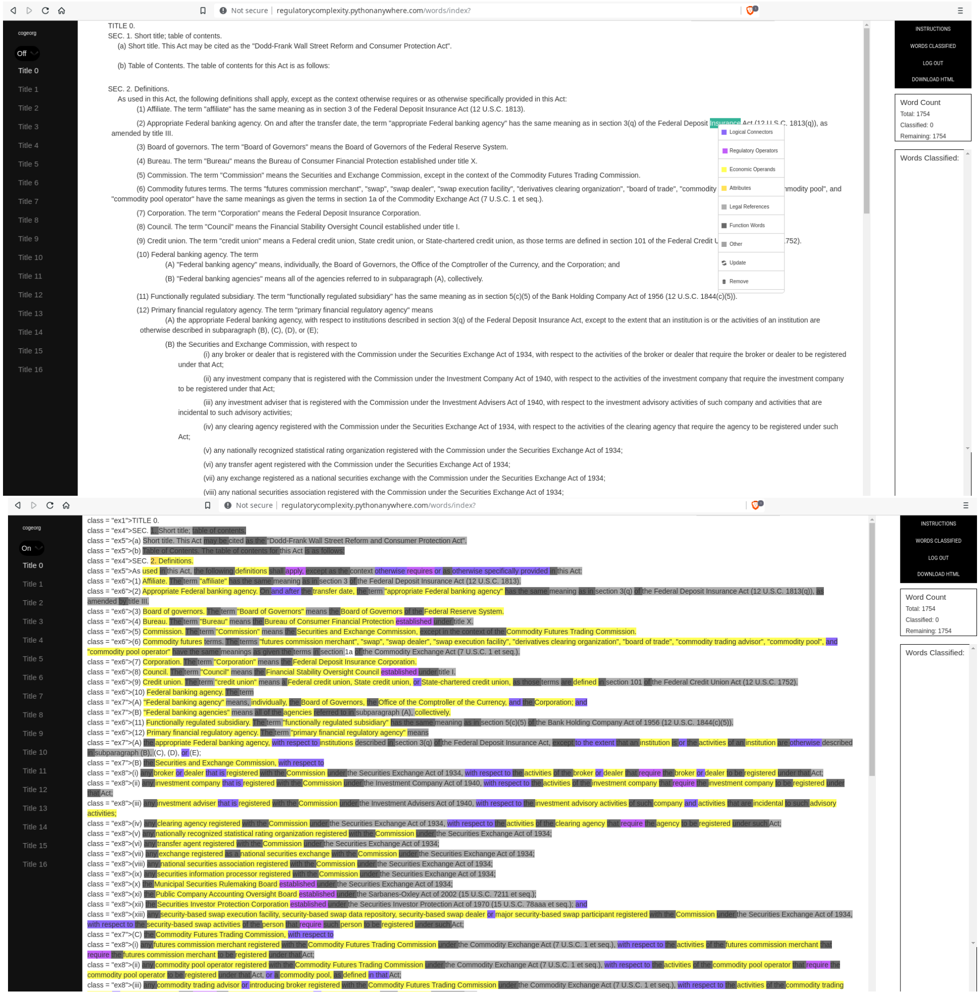


Figure 2: The Dashboard we developed to help us classify words in the Dodd-Frank Act as one of the following seven categories: Logical Connectors, Regulatory Operators, Economic Operands, Attributes, Legal References, Function Words, or Other. Top: The plain text of the Dodd-Frank Act. When highlighting a word or phrase, our dashboard displays a simple drop-down menu from which the category can be selected. The dashboard also provides some simple statistics on the right of the screen, and navigation on the left. Bottom: A mark-up of the Dodd-Frank Act when all words and phrases have been classified.

B Proof of Section 3

Proof of (6). For a given $E \in [a,b]$, total welfare is given by:

$$W_{a,b}(E) = \int_a^b [-\lambda(1-p)\min(x-E,0) - \delta E]dx \tag{14}$$
$$= -\lambda(1-p)\int_E^b (x-E)dx - \delta E(b-a) \tag{15}$$
$$= -\lambda(1-p)\frac{(b-E)^2}{2} - \delta E(b-a). \tag{16}$$

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:

$$W_{0,\bar{x}}(E^*_{0,\bar{x}}) + W_{\bar{x},1}(E^*_{\bar{x},1}) = \delta\bar{x}\left[\frac{\delta\bar{x}}{2\lambda(1-p)} - \bar{x}\right] + \delta(1-\bar{x})\left[\frac{\delta(1-\bar{x})}{2\lambda(1-p)} - 1\right] \tag{17}$$
$$= \delta\bar{x}(1-\bar{x})\frac{\lambda(1-p)-\delta}{\lambda(1-p)} - \frac{\delta}{2\lambda(1-p)}[\delta - 2\lambda(1-p)]. \tag{18}$$

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 $[0,1/I],[1/I,2/I]...[(I-1)/I,1]$. The $i+1$-th interval has a welfare of:

$$W_{i/I,(i+1)/I}(E^*_{i/I,(i+1)/I}) = \frac{\delta}{I}\left[\frac{\delta}{2I\lambda(1-p)} - \frac{i+1}{I}\right] \tag{19}$$
$$= \frac{\delta}{I^2}\left[\frac{\delta - 2\lambda(1-p)}{2\lambda(1-p)} - i\right]. \tag{20}$$

We use this expression to compute (8).

C 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. To compute the Halstead measures for each item we consider the code contained between two "ASSET_CLASS ==" (excluding this expression).

```
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;
```

D 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;
```