HAL open science

# Rules, Computation and Politics

Denis Merigoux, Marie Alauzen, Lilya Slimani

**To cite this version:**

Denis Merigoux, Marie Alauzen, Lilya Slimani. Rules, Computation and Politics. Journal of Cross-disciplinary Research in Computational Law, 2024, 1 (4), pp.23. hal-03712130v2

**HAL Id: hal-03712130**

**https://inria.hal.science/hal-03712130v2**

Submitted on 9 Dec 2022 (v2), last revised 12 Feb 2024 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

# Rules, Computation and Politics: Scrutinizing Unnoticed Programming Choices in French Housing Benefits

DENIS MERIGOUX, Inria, France

MARIE ALAUZEN, LISIS (UMR CNRS 9003), France and CSI (IMR CNRS I3 9217), France

LILYA SLIMANI, Inria, France

The article questions the translation of a particular legal statement, a rule of calculation of social rights, into a computer program, able to activate the rights of the concerned citizens. It does not adopt a theoretical perspective on the logic of law and computing, rather a realistic stance on contemporary welfare states, by studying the case of the calculation of housing benefit in France. Lacking access to CRISTAL, the source code of the calculation, we simulated the code base from the letter of the law and met with the writers of the housing law in the ministries to conduct a critical investigation of the source code. Through these interdisciplinary methods, we identified three types of unnoticed micro-choices made by developers when translating the law: imprecision, simplification and invisibilization. These methods also uncover significant sociological understanding of the ordinary writing of law and code in the administration: the absence of a synoptic point of view on a particular domain of the law, the non-pathological character of errors in published texts, and the prevalence of a frontier of automation in the division of bureaucratic work. These results from the explicitation of programming choices, lead us to plead for a re-specification in the field of legal informatics and a reorientation of the investigations in the field of the philosophy and the sociology of law.

Additional Key Words and Phrases: critical code studies, legal expert systems, regulation, social rights, sociology of law

## 1 INTRODUCTION

### 1.1 Technical Specifications and their Consequences

Modern administrative services use computer programs to automate the massive application of certain legal directives. One of the most emblematic examples of such automation lies in the computation of social benefits, distributed in many welfare states by government agencies. The program computes the amount of benefits owed by the government to the individual or household, depending on a long list of inputs characterizing it. The algorithm followed by the program is determined by a set of laws and regulations providing the legal basis for the welfare State.

The programs computing social benefits, implemented since the late 1970's in public administrations, are sometimes referred to as *legal expert systems*. They consist in a series of logic rules that effectively define a specification of a particular interpretation of the law, as described by Merigoux [2022b]. In a nutshell, the specification is a socio-technical *mediation* that bridges the gap between the intent of the legislator, more or less accurately conveyed by the text of the law, and the limited set of operations a computer is able to execute. In particular, the specification should effectively describe a computation, with determined inputs and outputs, and a series of computation steps between the two. As such, the specification is merely a projection of the law with a narrow application objective that does not capture the full descriptive power of the law, but rather seeks to disambiguate it completely for a specific purpose, usually at price of embedding several interpretations and tacit knowledge into the program.

While less powerful in collective meaning and interpretability than the text of the law, this computer specification has the benefit of increasing file processing efficiency and saving labor costs compared

to a multitude of civil servants using pocket calculators in their offices. At a second order of consequence, the centralization of the computation from the hands of the civil servants into a single computer program can help lower the level of *discretionary power* and improve sometimes imperfect law enforcement introduced by the human treatment of clerical administrative procedures[1].

Lately, some experts promoted by the OECD suggest there are advantages to using a novel approach in digital government services directly based on computer specifications, and called "rules as code" [Mohun and Roberts 2020]. In the case of social benefits in the US, Kennan and Soka [2022] write that the massive and computerized application of benefits law across the US has emphasized the small differences that existed in all the social programs enacted by the States and the federal government. These small differences regarding the definition of the staples of benefits law (income, family status, etc.), already detrimental to confused beneficiaries that sometimes give up benefits applications, are magnified by the computerized treatment of benefits eligibility. With centralization of eligibility screening, programmers of the legal expert systems have to deal with inconsistencies of legislation designed independently across branches of government. The "rules as code" doctrine then consists in promoting a transformation of the standards for writing legal rules, so that they can be directly transposed into a programming language and there is no gap between the legislator's intention and large-scale application. It is part of what Hildebrandt has labeled "code-driven law" [Hildebrandt 2020]. Some government agencies, like RegelSpraak in the Netherlands [Corsius et al. 2021], have already adopted at least partially this methodology. One of the problems is that RegelSpraak and related works feature collections of atomic computation rules, related to boxes of the tax forms themselves rather than the text of the law. This makes it harder to maintain a direct correspondence between the source code of the software and the text of the law.

### 1.2 Turning the Rules into Code: a Focus on Social Implications

While some authors, with a normative stance, call for an increased use of "rules as code" methodology into the very design and enactment process of legislation to reduce these inconsistencies, we first suggest a comprehensive investigation of the current computational complexity of turning legal rules into code. From a computer scientist's point of view, social benefits eligibility and computation software routinely exhibit buggy behavior, as laid out by Escher and Banovic [2020]. Some of these bugs come from a purely technical failure of the IT system, others are more interesting to us as they stem from the way the computer specification translates the intent of the law. In the latter, the term "bug" is not always appropriate as the unexpected behavior of the program does not always reflect an error, but rather a divergence about how the program *should* behave.

Indeed, although it should stem in all impartiality from the text of the law, the specification conveys the point of view of the program authors (and the administration that they work for), in multiple ways. Firstly, when the law is ambiguous, the administration responsible for writing and operating the computer program has to choose a single way in which the computation will be encoded, due to the rigidness of computer programs portrayed by Diver [2021]. The sources of ambiguity are diverse. Sometimes, the law is purposely ambiguous so as to give some leeway for its interpretation: for instance, alleviating sanctions for an indicted individual acting "in good faith". It is impossible to formalize what good faith is, as it is a notion heavily dependent on context; we shall not try to define its meaning in a computer program. Other times, the law is *inconsistently* drafted for programming and leaves open different and conflicting syntactic or semantic interpretations. In

[1] According to Lipsky [1980], public policies are made by street-level workers who regularly interact with citizens in the course of their jobs and that develop techniques to manage with rules and agency pronouncements.

this article, we will not expand on these ambiguities as they have been a staple of computational law studies for more than 60 years [Allen 1956].

Secondly, and this will be the core subject of this article, it is up to the writers of the program to define the *model of reality* that their computer program will act upon. Their work is precisely to "inscribe" this vision of the world in the technical content of the program. This vision of the functioning of technical objects echoes the sociological notion of "script". Akrich [1992] suggests that "like a film script, technical objects define a framework of action together with the actors and the space in which they are supposed to act." The interest of this notion is to underline that the behavior of technical objects, even the most sophisticated ones, does not always match the will of their designers, they have an "agency". Concretely, due to programming micro-choices, a form will ask from the recipient a certain amount of personal information required to assess their eligibility to the benefit, and compute the precise amount to which they are entitled to. Such personal information usually includes the income, family situation, dependent children, etc. Star [1990] has underlined the cost of surveillance that is borne by users when filling out the forms, the impossibility for the administration to guarantee that particular situations are effectively taken into account, and the inevitable alteration of relationships when a user makes a non-standard request[2]. Other sociologists have shown the personal information provided by the potential beneficiary is the key of their interaction with the State [Alauzen 2019; Dubois 2021; Weller 1999]: it should be detailed enough to fill the boxes of the administrative forms, but also backed by paperwork and sometimes very thorough investigations in users' homes that constitute the evidence of the reality of the information. For example, a period of the career of the potential beneficiary for which they cannot provide payslips might be rejected for consideration of the pension computation, leading to a lower overall pension. Because the programs determine the script that both users and administrations have to follow, we claim that programs have power in setting the relationship between the state and citizens, and their design around translating law in computation devices is a site of neglected discretionary power, or more exactly *sub-politics*, that should be placed under public scrutiny.

### 1.3 Case-study: Housing Allowances in France (1945-2022)

To scrutinize the politics of unnoticed programming choices in legal computation, we select the application for housing benefits in France. Our choice was informed both by the very broad nature of this benefit (millions of beneficiaries, dozens of billions of euros distributed) and the fact that its computation has become a public issue [Zerouala 2021][3].

Three categories of housing benefits were brought in the French welfare system (hereinafter referred to the contemporary National Family Allowance Fund as CNAF) throughout the second part of the 20$^{th}$ century. The first is the family housing allowance (hereinafter referred to as ALF), which was created in 1948[4] in order to compensate for the distortions in the housing sector caused by the Second World War and the increase in rents resulting from the law of September 1$^{st}$ 1945, enacted to rebuild housing and the nation. In the early 1970's, a new category of housing benefit was created to assist other vulnerable households which did not meet the eligibility conditions for the ALF, the

[2] "This is not the disenfranchised, which may at some point be 'targeted'; not the residual category not covered in present marketing taxonomies. This is that which is permanently escaping, subverting, but nevertheless in relationship with the standardized. It is not nonconformity, but heterogeneity." [Star 1990]

[3] Following the release of the pre-publication, we have been contacted by several investigative journalists.

[4] "Loi n° 48-1360 du 1$^{er}$ septembre 1948 portant modification et codification de la législation relative aux rapports des bailleurs et locataires ou occupants de locaux d'habitation ou à usage professionnel et instituant des allocations de logement."

social housing allowance[5] (hereinafter referred to as ALS). Finally, the third and last category, to which the bulk of this article is dedicated, is the personalized housing benefit (hereinafter referred to as APL) created in 1977 by the so-called "Barre" reform[6] (named after Raymond Barre, the minister who initiated it). The computation of the amount of these three allowances is complex and depends, on the geographical area in which the accommodation is located, the amount of the rent or the monthly loan repayments, the size of the family and its resources.

The management of housing allowances and, more generally, family benefits is carried out by the departmental family benefits offices of the CNAF. They are located throughout the country and interact with users daily. These offices were created by the decree of October 4, 1945, the Article 19 of which stating that: "The management of family benefits is ensured by family benefits offices whose district and headquarters are determined by decree of the Minister of Labor and Social Security, taking into account the territorial districts of the primary social security funds"[7]. Additionally, personal housing assistance and removal grants are financed by the national housing assistance fund[8], which provides the legal framework and standardizes the procedures. While the CNAF has invested in automation since its creation in 1945, in 1997 it introduced the CRISTAL[9] information system to handle the increasing complexity of the legislation and regulations for all its benefits, including housing benefits, and compute allowances [Kounowski 2002].

The logic followed by the French State for the creation of the APL, that of a generalized benefit whose amount depends on the ability to pay the rent, differs considerably from previous policies [Calcoen and Cornuel 2001]. The APL pursues more of an overall objective of a generalized personal benefit, which also conflicts with previous policies of subsidies to property-owners, aimed more at investment in real estate. Public intervention on these issues increased at the end of the 1980's and beginning of the 1990's in order to extend the scope of this benefit to all rental housing, but also in a desire for social equity towards more beneficiaries. If the intent behind housing benefits that transpires from all the policies taken during the 20$^{th}$ century appears to tend more towards social action, the trend has been reversed since the beginning of the 21$^{st}$ century, especially during president Macron's first five-year term (2017-2022).

Several reforms have, in fact, resulted in the reduction of the housing benefits and have contributed to the public concern. The first of these reforms, introduced by a decree, was materialized by a flat-rate reduction of 5 euros for all recipients of personalized housing benefit[10]. Another reform was introduced by the decree of February 27, 2018, which provides for a decrease in personalized housing assistance correlating with the reduction in solidarity rent for households that benefit from it[11]. The latter was introduced as part of the 2018 budget legislative package and involves a cut in the benefit paid by the State but borne by social landlords[12]. Thus, this does not appear to be a reduction in the amount of benefits, but it is nevertheless a source of savings for public finances and a change

[5] "Loi n° 71-582 du 16 juillet 1971 relative à l'allocation de logement."
[6] "Loi n°77-1 du 3 janvier 1977 relative au maintien des aides publiques à l'investissement malgré l'institution de l'aide personnalisée au logement." This law gave rise to the classic work of Bourdieu [2005] on the market of the individual house.
[7] "Ordonnance du 4 octobre 1945, Article 19 : 'La gestion des prestations familiales est assurée par des caisses d'allocations familiales dont la circonscription et le siège sont fixés par arrêté du ministre du travail et de la sécurité sociale, compte tenu des circonscriptions territoriales des caisses primaires de sécurité sociale.'"
[8] Articles L811-1 and following of the Construction and Housing Code.
[9] Acronym for "Conception Relationnelle Intégrée du Système de Traitement des Allocations".
[10] "Décret n° 2017-1413 du 28 septembre 2017 relatif aux aides personnelles au logement et au seuil de versement des allocations de logement."
[11] "Décret n° 2018-136 du 27 février 2018 relatif à la baisse de l'aide personnalisée au logement dans le cadre du dispositif de réduction de loyer de solidarité."
[12] "Loi n° 2017-1837 du 30 décembre 2017 de finances pour 2018, Art. 126."

of philosophy, which is transferring the "social burden" to the private sector. Later, in 2019, a new method of computing personal housing benefits was introduced. The updated computation implies that income and resources are taken into account "in real time", over the last twelve months and no longer over a period of time two years earlier[13]. Numerous adjustments were necessary following the adoption of these decrees, but, beyond the succession of technical rules, the political intention that has not escaped observers is a significant reduction in personalized housing benefit [Cour des comptes 2020]. In sociological terms, it is a process of "sub-political economization", that presents itself as economic, but takes the place of politics through its consequences [Linhardt and Muniesa 2011].

## 2 DATA AND METHODS: FIXING THE ABSENCE OF AN EXPLOITABLE SOURCE CODE BASE

### 2.1 Source Code Investigation

To study the sub-politics that comes into the computation of housing benefits in France, we will dive into the technical intricacies of one of the legal expert systems used by the French administration to power the financial redistributions operated at the national level. Computer systems taking automated administrative decisions are under quite heavy academic scrutiny in two main circumstances: in case of scandals or affairs, or, more recently, when the source code is open. For instance, in France, the introduction of "Parcoursup", a matchmaking algorithm between prospective first-year students and universities launched a collective process of investigation into the allocation of training courses, in which economics, sociology of education and computational sciences were involved [Becker et al. 2020; Frouillou et al. 2019]. Such critical investigation indeed requires a combination of skills between social sciences, law and computer science.

Apart from these situations in which the public investigates administrative software, the main obstacle for critical investigation has been the access to the source code of these legal expert systems. Since 2016 in France[14], the source code of legal expert systems involved in automated administration decision making is subject to the same access rights as administrative documents. Nevertheless, the publication of the source code is operated independently by each administration. While a lot of source code has been released on code.gouv.fr [Guerry 2019], for computer scientists the material is not always fit for reuse, exploitation or even investigation. Indeed, the legal expert systems operated by the French administration often use legacy technologies and programming languages like the historic language for business operations in mainframes, COBOL. Some programs are even written in a custom programming language used only by the administration [Merigoux et al. 2021b]. Because they date from times when writing, verification and collaboration practices were not the same as today, these legacy or custom programming languages sometimes lack the infrastructure necessary to run programs written with them on modern machines: compilers, interpreters, etc. The published source code also rarely comes with the documentation[15] about how to feed the program input, what does the program even compute exactly, or how to interpret the administrative terms inside the names of functions and variables. For resolute computer scientists, the absence of documentation may be compensated by a significant retro-engineering effort, which can also prove impossible to achieve if some key information was withheld by the administration during publication. Hence, even if the source code of the legal expert system is published, it is

[13] "Décret n° 2019-1574 du 30 décembre 2019 relatif aux ressources prises en compte pour le calcul des aides personnelles au logement."

[14] "Loi n° 2016-1321 du 7 octobre 2016 pour une République numérique."

[15] Guidelines for high-quality software artifacts documentation are for example released by the Association for Computing Machinery [2020].

often unworkable for a critical investigation of the underlying algorithm, for instance to check its faithfulness to the legislation it is supposed to follow.

This situation holds true for the case study of this article, concerning the computation of French housing benefits. Even though they are distributed locally by a network of social benefits agencies, the rules that specify the eligibility and amount are enacted and translated into code for the CNAF's IT system, CRISTAL, at the national level. When we asked for the source code of the housing benefits computation in October 2020, the CNAF replied by physically mailing us a CD-ROM containing a set of COBOL files similar to the ones obtained two years earlier by journalists [Berne 2018]. These COBOL files, lacking any documentation about how to compile or run them, appear to be generated by a Computer-Aided Software Engineering (CASE) tool, whose high level sources have not been published. In other terms, what the CNAF published is merely the low-level source code, not meant to be read or edited by humans, while the real high-level source edited through a CASE tool is still unavailable to the public.

Moreover, we received in November 2021 during subsequent discussion with the CNAF another excerpt from the source of the housing benefits computation. This excerpt was written using Oracle's proprietary Oracle Intelligent Advisor (OIA) programming language, as it appears that the CNAF had recently undertaken a rewrite of the housing benefits computation algorithm using this new technology, as a part of a grand scheme to modernize the CRISTAL system. We asked for the rest of the source code written in OIA but got no answer[16].

### 2.2 Simulation of the Codebase

In the absence of an exploitable source code base, we have developed a working implementation of the housing benefits computation so that we could study it instead of the "official" one. This method of simulation is common in critical code studies. We have used the Catala programming language and methodology already empirically tested for the tax code [Huttner and Merigoux 2022; Merigoux et al. 2021a] to translate the housing benefits computation into computer code directly from the text of the law. This programming methodology can be related to the "isomorphism" approach [Bench-Capon and Coenen 1992].

Concretely, we have spent more than 70 hours doing interdisciplinary *pair-programming* as a lawyer-programmer duo to produce more than 7,000 lines of computer code[17] and 6,000 lines of legislative text, literally constituting a computer program able to compute the eligibility and amount of housing benefits entitled to a household described by a computer form. This form, as well as the computation rules, have been extracted from a corpus of legislative and regulatory provisions constituted after an extensive legal search of everything related to housing benefits on the French legal database (legifrance.gouv.fr). Once the corpus had been constituted, the translation to code

[16]The rewriting of this legal expert system, whose new source code is an administrative document, using a proprietary technology (Oracle Intelligent Advisor) is problematic in light of article L300-4 of the French code of relationship between the public and the administration: "Every electronic publication done in application of the present book has to be done in an open standard, easily reusable and exploitable by an automated processing system". If the programming language in itself cannot be copyrighted [Pellegrini and Canevet 2013], the compiler and associated tooling could be withheld by Oracle. This severely limits the reproducibility and usability of the published source code because one would have to retro-engineer the compiler and associated tooling to execute the new CNAF housing benefits source code.

[17]The ratio of lines of code per hour of programming was very high because of several factors. First, the code contains a lot of copy-pasting that mirrors the duplications in the law. Second, a lot of boilerplate code comes from the translation of huge tables of values, contained in an executive order assigning threshold amounts of benefits. The logic of the table had to be expressed in terms of conditionals and pattern matching in a very repetitive fashion. Third, the lead programmer is one of the two authors of the programming language [Huttner and Merigoux 2022; Merigoux et al. 2021a], this scenario ensures a maximum programming productivity.

was done in the order of the legislative texts, following the hierarchy of rules from high to low. The resulting source code is published under the Apache license on GitHub and archived on Software Heritage. If printed as a textual document, a snapshot of the codebase would result in over 280 pages.

During the process, we have become aware of the inventiveness and effort required to translate the law. It is not self-evident, but challenging, there are always several ways to do it. Therefore, we have recorded all the instances where we realized that we have to make a micro-choice in the textual interpretation of the legal framework that may affect the outcome of the computation or the attitude of the beneficiary. We then grouped these choices into categories. We present below three of these categories as detailed case studies: imprecision, simplification and invisibilisation. They were selected because each explicits a kind of unnoticed decision programmers make.

### 2.3 Coupled with a Sociological Inquiry into the Administration

Bringing sociological methods to critical code studies, we carefully looked for CNAF forms (both paper and online) and contacted the ministries responsible for writing the texts that govern the computation of housing allowances, in order to verify that the simulation of the code is as close as possible to the letter of the law. We conducted four interviews, with ten people in 2022, and exchanged a dozen emails in long threads with multiple documents. We then realized that our Catala simulation probably does not tell us much about a program like CRISTAL, designed in the 1990's with technical specifications and constantly updated since then. Yet, by bringing such a "boundary-object" [Star 2010] we have asked precise questions about the administrative way of writing the law and regulating public agencies, and we have learned a lot about law in practice.

Through these exchanges, public agents helped us to find errors in our source code as we sent them the list of inconsistencies and typos that we had found in published texts. The collaboration has gone far, as we have even reviewed drafts of decrees, and wrote an expert report on a computational problem concerning housing benefits computation that is focusing public attention [Merigoux 2022a]. The interaction allowed us to understand the ordinary conditions of the production of this type of rules (formal and informal constraints, justifications and incertitudes, negotiating sequences, the intertextual construction of legal rules…), to which we will return later.

## 3 MICRO-CHOICES EXPLICITATION

### 3.1 "Should we Ask Couples for the Date of Conception?" Imprecision at Stake

The first case study chosen to expose the type of choices involved in translating the law into a program concerns the computation of the moving allowance, which depends on the state of the pregnancy of the potential beneficiary. This social benefit is set out in Article D823-20 of the Construction and Housing Code which provides:

> The moving allowance is awarded to persons or households *with at least three children born or to be born* who move into a new housing entitled to one of the personal housing allowances *during a period between the first day of the calendar month following the third month of pregnancy in respect of a child of rank three or more and the last day of the month preceding that in which the child reaches his or her second birthday* […][18]

[18] "Code de la construction et de l'habitation, Article D823-20 : 'La prime de déménagement est attribuée aux personnes ou aux ménages ayant à charge au moins trois enfants nés ou à naître et qui s'installent dans un nouveau logement ouvrant droit à l'une des aides personnelles au logement au cours d'une période comprise entre le premier jour du mois civil suivant

This mode of determining the temporality of personal rights is widespread in law, but it does not fit well with computational logic. Indeed, this article is complex for programmers in that it makes the eligibility for a social benefit (in this case the moving allowance) dependent on an uncertain event or at least of uncertain timing, namely the birth of a child.

The main issue here is to determine which date has to be taken into account for the start of the pregnancy in order to ascertain when that period of three months effectively starts to be able to compute the allowance. Two options can be envisioned for the computation: the date of the conception of the child or the date considered to be the beginning of the pregnancy. However, the search for this kind of data may be deemed too intrusive or non-relevant from the point of view of the beneficiary completing the form and may lead to non-take up.

Another issue was the computation of periods of time, specifically here, months[19]. It is therefore difficult to establish a time frame for the first day of the civil month of the third month of pregnancy with precision. In other words, this article does not really provide a clear starting date for the time period it sets, thus, making it challenging to incorporate into the program the most accurate time and duration assessment method possible.

Beyond these technical considerations, there is also the question of the usefulness of such personal data for the computation of the personalized housing benefit. The search for these elements is intrusive in the private life of the potential beneficiary. It is therefore essential to question the need for such information as the choices made at this stage will have a direct impact on the questions asked in the form and on the user who will proceed with an application.

The question then becomes: what does the administration's program do? Will the programmer look for the easiest information to code, even if that information turns out to be intrusive and a potential deterrent to claiming a right? Or will she look for alternative solutions to preserve privacy and improve access to rights? Without access to the source code, we do not know precisely how CRISTAL computes the pregnancy condition. Nevertheless, given the official form we have had access[20], we deduce that the program uses medical data named "the presumed date of the beginning of the pregnancy" that medical professionals determine to plan mandatory examinations (such as ultrasounds), preparatory interviews and childbirth and parenthood preparation sessions with midwives. In other words, for the information system's convenience, the housing allowance computation system asks users to provide medical data filled in another form, for another purpose[21].

In addition to making these kinds of choices explicit, simulation in Catala allowed us to experiment with alternative paths that show that it could be otherwise. Hence, translated in Catala, the article D823-20 is as follows[22] :

le troisième mois de grossesse au titre d'un enfant de rang trois ou plus et le dernier jour du mois précédant celui au cours duquel cet enfant atteint son deuxième anniversaire [...]'"

[19]The concept of time is complex to implement through a computer language as there are various ways to measure it. Most computers count the number of milliseconds elapsed since Jan. 1st, 1970 for their time computations, but French law implicitly relies on computations using the Gregorian calendar, which themselves can be ambiguous. For instance, what is "Mar. 31st + 1 month"? Is it May 1st or Apr. 3? Programmers always have to specify it.

[20]"CERFA 50040#06, Premier examen médical prénatal."

[21]Further research is needed to determine compliance status of this with respect to GDPR. However this pre-dates GDPR and the CNAF's CRISTAL has been regulated by the French National Commission on Computer Science and Freedoms (CNIL) for a long time, so it is likely that this use of medical data for housing allowance was planned and approved.

[22]All code excerpts presented in this paper have been translated from French to English; indeed Catala programs are written to match the natural language of the legislative source.

construction_code_regulations.catala_en

```
rule condition_moving_period under condition
  (match household.birth_date_third_child_or_last_if_more with pattern
  -- LessThanThreeChildren: false
  -- MoreThanThreeChildren of birth_date_or_pregnancy:
    (match birth_date_or_pregnancy with pattern
    -- BeforeFirstDayOfThirdMonthPregnancy: false
    -- AfterFirstDayOfThirdMonthPregnancy: true
    -- BirthDate of birth_date:
      current_date <=@
        ((first_day_of_month of (birth_date +@ 2 year))) +@ (-1 day)))
consequence filled
```

We have chosen to leave it up to the user to determine whether his or her request is actually made after the first day of the calendar month of the third month of pregnancy, via the use of an input variable taking values into a predetermined enumeration. Thus, no unnecessary data is collected, the data required to determine eligibility for the removal allowance is verified and the form remains neutral. This solution results from the breakdown of the various possible hypotheses:

- The household has less than 3 children and is therefore not eligible;
- The household has 3 or more children and the last child is already born: the period runs from the child's birth to the last day of the month before the month of the date of the second birthday;
- The household has 3 or more children and the last child is still to be born: it is then up to the user to ensure that the application is made after the first day of the calendar month of the third month of pregnancy.

This decision was taken with a view of protecting the privacy of beneficiaries while ensuring compliance with the law and accuracy of computation. Privacy is protected in French law at several levels: first by law since 1970[23], then by the Constitution since 1999[24], but also at the European level by Article 8 of the European Convention for the Protection of Human Rights and Fundamental Freedoms. By translating this law in such a way that there is no need to process such information in order to determine eligibility, the protection of the privacy of the potential recipient is ensured (to the extent of any information necessary and useful for the determination of the eligibility of the recipient).

The decision to simplify the information required in this case as much as possible also had a significant effect on the translation of the text into code. The relevant part of the program was considerably simplified as we removed some of the complex date computations.

The question of how to compute the duration set out in the law indicates that programming choices have consequences for the input forms that users fill out. They are directly involved in their more or less privacy-friendly nature and have heavy implications for the user. First, the most convenient programming choices for developers may breach the privacy of the user that has to supply extensive personal information that is not *per se* required to apply the law. Second, they might create form boxes impossible to fill by users whose situation had not been planned, making the form harder to

[23] "Everyone has the right to privacy", Civil Code art.9, in force since July 19 1970.
[24] "Conseil Constitutionnel, 23 juillet 1999."

fill or reducing the interpretation leeway intended by the legislator and consequently reducing the use of rights. And all this, because of a question of precision of the duration of a parameter entitling to a moving grant, while this parameter can itself only be determined with uncertainty.

### 3.2 "Are There Too Many Labels in the Law?" The Temptation to Simplify Categories

The second micro-choice concerns Article R821-3 of the Construction and Housing Code providing:

> In the event of legal or *de facto* separation of the spouses *resulting in the creation of two separate households* and the occupation of two main residences, as determined by the paying agency at the time of entitlement or at the beginning of the payment period, *personal housing assistance may be granted to each spouse*[25].

In this case, it is not so much of an imprecision of the law but rather a situation that does not appear to be taken into consideration in other articles of law (especially those determining the family situation) or by the administration when creating the forms used by the beneficiaries to apply to these housing benefits. "*de facto* separation of the spouses" is therefore a category that exists only in this legal device and is apparently confusing with other family situation categories ("single"), which programmers may tend to overlook.

On the CNAF online form (accessed on June 21st, 2022), there is only one possible choice: "You live as a couple: yes or no". We therefore deduce that the computer program does not respect the letter of the law — because it is concerned with not multiplying the questions, or because it has forgotten or is not aware of this category. Yet, it is a situation that can arise for a couple. They may occupy two separate housings and are entitled to receive this benefit individually, although they might still be legally married and appear as one household. Trickier: if a separated-but-still-legally-married person starts cohabiting with a new partner, then they should file as a couple with the new partner. Accuracy of the form that program makes and couples complete is of utmost importance here since any lack of precision could prevent a potential beneficiary from accessing rights to which they would otherwise be entitled to.

Again, the simulation in Catala shows that it could be otherwise. To ensure adequacy with R821-3, the review of existing categories for the "family status" provided in Article L822-1 was required[26]. This was translated in Catala as follows:

construction_code_law.catala_en

```
declaration enumeration FamilyStatus:
  -- Single
  -- Married content date # Wedding date, used by L841-1
  -- CivilPartnership
  -- FreeUnion
  -- SingleDeFactoSeparated # See R821-3
  -- FreeUnionWithDeFactoSeparated # See R821-3
```

[25] "Code de la construction et de l'habitation, Article R821-3 : 'En cas de séparation, légale ou de fait, des conjoints entraînant la création de deux foyers distincts et l'occupation de deux résidences principales constatées par l'organisme payeur lors de l'ouverture du droit ou au début de la période de paiement, une aide personnelle au logement peut être accordée à chacun des conjoints.'"

[26] Article L822-1 of the Construction and Housing Code: "The provisions of this book relating to the beneficiary, the principal residence or the taking into account of resources *applicable to the spouse are applicable, under the same conditions, to the partner linked by a civil solidarity pact or to the cohabitant.*"

Instead of creating a whole new category, we assessed the different possible situations which could emerge from a case where the couple is *de facto* separated. We came to the conclusion that each person *de facto* separated could either be single or in another relationship and cohabiting with the person. We then used two pre-existing categories: "single" and "cohabiting partners" to fit this new scenario giving birth to two new variables: "*de facto* separated and single" and "*de facto* separated and cohabiting". Because they are attached to the same enumeration in Catala, the programming language will force the programmer to review those two new cases every time the family status is used in the code of the program. Here, we use a technical feature of the programming language (here, its type system) to ensure consistent application of a legal subtlety across the codebase.

Revealing the family status of the user is necessary for the housing benefits computation as it allows the administration to determine the exact number of people occupying the accommodation. Being as precise as possible and not to simplify legal categories in form is thus mandatory. Catala allows us to determine all the hypotheses to have the most representation, so that when the user completes the form, she will not hesitate between categories and have a choice actually consistent with the reality of her situation. Indeed, she might not complete an application for housing benefits because she is not aware that both of them have access to it or giving up because none of the previous categories fit their personal situation which could lead her to believe that she does not have the right to such housing benefits. This reasoning allows for simplified access to housing benefits for this particular case for certain beneficiaries who might otherwise suffer from a lack of information or knowledge in this regard.

### 3.3 "Does the Computation Always Benefit the User?" On Corner Cases Invisibilization

The third micro-choice explicitation we identified with the simulation method focuses on the processing of the different housing benefits when dealing with an application made by a potential beneficiary. Article L841-2 of the Construction and Housing code provides:

> People who do not benefit from the family housing allowance or the personalized housing allowance can claim the social housing allowance[27].

Although quite understandable at first glance, the phrasing of this article leaves the door open for interpretation.

One may claim that it sets a hierarchy between these different allowances as it prioritizes the APL over the ALF and the ALS. The Ministry of Housing, we questioned on this point, has confirmed to us by email this is the interpretation adopted by CNAF[28]. According to them: if a household is eligible to the APL, they will benefit from it. If they are not, but eligible to the ALF, then they will benefit from the ALF, and finally, if they are not eligible either to the APL or the ALF, they will benefit from the ALS.

A slightly different interpretation may rest on the *de facto* distinction between being eligible for an allowance and benefiting from it, *i.e.* between meeting all the requirements set out to access this benefit and effectively receiving it. The legislator's choice to create a rule which relies on the actual payment of the allowance to the beneficiary rather than on eligibility seems to reflect a desire to prevent beneficiaries from accumulating allowances, rather than the establishment of a decree of priority in the treatment of housing allowances.

[27] "Code de la construction et de l'habitation, Article L841-2 : 'Les personnes ne bénéficiant pas de l'allocation de logement familiale ou de l'aide personnalisée au logement peuvent prétendre au bénéfice de l'allocation de logement sociale.'"
[28] Email exchange with DGALN/DHUP/FE4, May 30, 2022.

Questioning the initial interpretation came from the mathematical realization that the computation of these allowances differed, which implies that it may sometimes be more profitable for a household to receive one allowance than another. Yet, if a person is eligible for the APL, they will receive the APL, regardless of whether they are eligible for another benefit that is more advantageous. It then proceeds that the interpretation of the administration might lead to financially hurting those eligible for both ALF/ASL and APL, and for whom the amount of ALF or ALS is higher than the amount of APL. This situation appears to have been overlooked by the administration, as a careful inspection of the computation formulae for the ALS/ALF and the APL leads to observing that in most cases, the amount of APL is indeed higher than the amount of ALS/ALF for the same household. However, these computation formulae have *corner cases*, specific situations where the actual result might deviate from the intended outcome. Corner cases are a notion with which programmers are familiar, as these are the cases that consume most of their attention and time when they write computer code.

We provide such a corner case for which the ALS/ALF amount is higher than the APL amount, in Appendix A[29]. This corner case corresponds to a household an elderly couple living in communal housing, one of them working for the minimum wage, the other having no income at all. We do not know how many real households this corner case corresponds to, as we do not have access to the CNAF's database[30]. Nevertheless, this corner case intuitively sounds realistic ; the difficulty to find such a case in reality will probably come from the relatively low rent that the couple pays (360 € per month), although they live outside of a big agglomeration subject to rent inflation. For this couple, the financial loss of taking the APL instead of the ALS/ALF is about 120 € a year, but would increase with a lower rent due to the nature of the formulae for housing benefit computation. Because we do not have access to CRISTAL, we could not test whether the system actually prioritizes the APL over the ALS/ALF on our corner case. We can only suppose that it is the case, based on our exchanges with the Ministry of Housing.

When completing the online form on the CNAF website, there is no mention of a possibility for the beneficiary to choose which benefit they are applying to. The CNAF decides for them, and this decision seems to prioritize the APL, sometimes at the expense of the beneficiaries' access to rights. The divergent interpretation of a rule whose initial intention may seem clear, seen from a certain perspective, can have a major impact on beneficiaries' access to benefits. Yet, one of the fundamental principles of social protection that we were reminded directly in the ministries is that the purpose of these measures is always to prioritize the beneficiary, compared to the administration or any other entity. While we could not find any hard legal backing for this principle, it is consistent with the historial construction of the French social security [Palier 2005]. In this case, it seems to us that there is an obvious contradiction between the principle and the interpretative choice.

In order to testify that it could be otherwise, for our simulation in Catala, we compute both the amount of ALS/ALF and APL for households that are eligible to both, and attribute the benefit with the maximum amount. This solution is clearly more complex technically, both in terms of number of lines of code, but also performance-wise. Indeed, this single choice is likely to double the computation time for each household. Re-running a computation with different parameters or tweaks to compare outputs is a practice sometimes called "multiple liquidations" by the administration developers we met.

[29] A necessary preliminary explanation on the computation of these allowances is also provided in Appendix A.
[30] As this database contains a lot of private information about millions of French households, its access by researchers is subject to a heavy procedure under the Committee for Statistical Secret.

## 4 SOCIOLOGICAL PERSPECTIVES

At this point, we do not know how CRISTAL computes social benefits, however our scrutiny from critical code studies has highlighted unnoticed programming choices, influenced by the writing of the law. Temptations of imprecision, simplification and invisibilisation can significantly affect the application of a piece of legislation having a big financial impact on its millions of beneficiaries. These programming micro-choices have certainly been encountered by CRISTAL's programmers, but also by many others when writing legal expert systems.

The method of transcribing French housing law line by line also taught us that there were some "mistakes" in the writing of the law. In this section, we propose to examine these mistakes in a sociological perspective, in the light of interviews conducted and email exchanges with the ministry agents responsible for writing these legal texts, and ruling the CNAF.

### 4.1 "Unscrewing" the Legislator in Writing Practices

The meetings with the agents of the French Ministries immediately confirmed one of the fundamental lessons of ethnomethodology studies in the field of law: there is *no legislator with a synoptic view of the law* [Dupret and Ferrie 2008; Latour 2010]. Even more, in the office responsible for personal housing assistance, there is not even a single document listing all the texts framing the personal right to housing. We put together this document as we were simulating the CRISTAL code and sent it to them[31]. So how do they manage it? When they have to modify or write housing law texts, the agents "go hunting for legal texts"[32] and do not always find the texts they need (especially when the administrative memos are old and there have been many versions). Moreover, our literal transcription of the law has shown us that a certain number of texts were never published[33]. The agents of the ministries confirmed to us that the information they contain still arrives in the administrations concerned by means of instruction documents and practical guides. This is not to say that in other places and in other areas of the law this synoptic view does not exist, but it is probably only temporary, partial and extremely difficult to build and maintain. As Latour [2010] has shown, the normal mode of existence of law is intertextuality, not a broad overview. We still need to "unscrew" [Callon et al. 1981] our modern fictions – here the Legislator – to grasp law in action.

The absence of this holistic view of the area of law for which the ministry agents are responsible is not due to failure or malpractice. It is the result of the *administrative organization* of the Ministry. The agents who take a position in the office responsible for the regulation of housing subsidies inherit Excel tables with complex rates, values and formulae. Their task is to update these values annually or quarterly in the table and in the related regulatory texts. The ability to write legal texts

[31] "Par ailleurs vous nous avez indiqué avoir listé l'ensemble des textes relatifs aux APL. Nous sommes preneurs de votre listing", Email exchange with DGALN/DHUP/FE4, May 25, 2022.

[32] Interview with DSS/AFPFAT/2B, June 8, 2022.

[33] This is the case for another benefit, the solidarity benefit for elderly people ("Allocation de Solidarité pour Personnes Agées", ASPA). The benefit is instituted by article L815-1 of the Social Security Code, and its maximal amount is set by article D815-1 of the same code. D815-1 is a decree that is supposed to be renewed each year, as the maximal amount for ASPA is indexed on the inflation. However, the latest version of D815-1 we could find dates from 2018, and we could not find any decree or order updating the values since then. We do know that the value is effectively updated each year, because we found yearly internal memos of the pensions agency (CNAV) managing the ASPA indicating the inflated values. We have sought comment about this situation to the relevant office and have been told that the decrees or orders are not necessary because the published rules for updating with inflation are sufficient. However we ourselves had trouble reproducing the computation from the rules, because correctly interpreting them required implicit knowledge about the publication calendar of administrative statistical series. The ASPA amount is needed for the housing benefits computation through article R823-4 of the Construction and Housing Code.

is a skill they acquire through experience [Torny 2005]. Positions in the Ministry of Housing are traditionally held by engineers[34]. During their education, they do not necessarily have law courses, and may not even be familiar with social protection regulations, the role nor history of the welfare state in France. However, the texts for which they are responsible take part in an advanced division of labor in the writing of the law in society and bureaucratic specialization, according to which the formatting of regulatory texts relies on a "sub-administration" Martinais [2010]. This configuration ensures a great deal of autonomy for subordinate agents in central administrations while opening up margins of maneuver [Page 2003].

Consequently, when academically-trained lawyers read the texts these civil engineers agents write, they figure out that there is no high concept of "Law", prompt to hermeneutics, but a rather complex activity. Regulation is carried out most of the time with no reference to the general principles of law, with little care for coherence and readability… Indeed, regulation may also seem to be *opaque to lawyers* (see, for instance, Appendix A about $\mathrm{ALS} = \mathrm{K} \times (\mathrm{L} + \mathrm{C} - \mathrm{L0})$ in the Article D842-15 of the Construction and Housing Code).

### 4.2 The Normal and the Pathological of Regulation at Work

During meetings with the agents who write housing law, we confirmed that the ordinary mode of production of legal rules and computer programs is structured by *current events*. Current events can be both legal (a finance law in which a revision must be included) or political (an announcement by the president of the individualization of a benefit formerly attributed to the household as a whole, or an unexpected parliamentary amendment). They shape the writing process, which then proceeds through multiple layers of proof-reading by the hierarchy, other administrations concerned, and experts. Due to the technical nature of the texts, and in order to save time, the agents use a lot of copy-pasting, changing only the dates or the referenced values. However, this practice of reproducing texts does not prevent typos and inconsistencies.

During our sessions when we were programming the housing benefits from the letter of the law, we found several errors. Indeed, while the law describes the general principles for housing benefits eligibility and computation, the values and parameters are detailed in a separate executive order[35]. The executive order contained several typos that affected some parameters of the computation (for instance "2987.27" written instead of "287.27" in article 33), and missed a key listing of parameters for a special case. Overall, three articles (18, 31 and 33) contained at least an error. When we reported the first of these errors to the ministries concerned, we were quietly told that "There are plenty of errors in the social security code"[36]. In fact, we were not the only ones to have detected them; some had already been pointed out to them by partner administrations, or agents had become aware of them while writing other texts. The *collaborative writing of the law* reflects a logic of self-regulation through the participation of the stakeholders, rather than a logic of control and inspection of administrative practices [Torny 2011]. As in the process of collaborative writing of computer programs, the correction of errors in regulatory texts is part of a normal process. In any case, the regulation must be maintained (*e.g.* every quarter some parameters must be updated for inflation), so errors are integrated into the work cycle of the ministry and do not trigger an alarm for the agents. They serenely modified the law in a new executive order published a few weeks after our exchanges, and reciprocally helped us find errors in our Catala simulation.

[34]From the State Civil Engineering Corps.
[35]"Arrêté du 27 septembre 2019 relatif au calcul des aides personnelles au logement et de la prime de déménagement."
[36]Interview with DSS/AFPFAT/2B, June 8, 2022.

We learned from these exchanges that rewriting is a part of the social processes of *maintenance of the law*, in which the regulatory texts are inscribed. The whole legal edifice does not collapse on the basis of an error in a computation parameter, however big it may be. Why? Because, on the one hand, CRISTAL's programmers do not re-encode directly from the text of the article each time, they just change a rate, a parameter and thus modify the program by small adjustments. Moreover, the thought that the errors are not pathological is confirmed by the complaint letters that the Ministry of Housing sometimes receives from users who dispute the computation of their situations by the CNAF. They manually redo the computations and conclude that "the computation is never a problem, the issue is the resources taken into account at the beginning"[37]; in other words, aspects relating to the readability of the form and the quality of the data provided by the users, not to the law governing the computation. That is the reason why errors in the publication of a legal text are not seen as pathological, resulting from a lack of professional conscience or understanding of the law of the agents.

Our simulation procedure does, however, highlight a point of vigilance. We noticed when reading the CNAF forms that there were simplifications and that not all situations governed by law could be declared (Section 3.2), but also that the interpretations of the law taken into account in the computation might not always benefit the users, thus contradicting the logic of social protection (Section 3.3). These types of issues may not be resolved in the same way and, for example, simplifications of the law may create behind-the-scene frictions with the CNAF. It is also possible that the problematization process of housing subsidies politicizes certain errors and redefines what is pathological. For example, journalists and associations are trying to impose the idea that the impossibility of splitting the housing benefits in the case of alternating custody, due to the technical debt of CRISTAL, is a public problem.

### 4.3 An Automation Frontier in the Bureaucratic Division of Labor

The exercise of simulating the CRISTAL code for calculating housing benefits opened up a space for dialogue with the administration. We have emphasized crucial elements about the division of bureaucratic labor, beyond the normal and the pathological. One of the first things we were told by the head of the office of family and housing benefits at the Ministry of Health and Solidarity is that those who write the law "believe too much in the performativity of the law"[38]. It means, in this context, that they deem the law is sufficient to bring about an alignment of the information systems, of the practices of the agents at the counters and of the whole bureaucratic process of housing benefits[39].

In this case, this belief is based on two phenomena. The first is an *automation frontier* in the division of bureaucratic labor, which has to do with the training of French civil servants and the self-regulation logic mentioned above. Those who write the law in the ministries are not (or do not consider themselves) able to read the source code of information systems to check whether they respect the letter of the law and have been correctly updated. The few among them who would like to access it in order to conduct impact studies of a regulatory change for example, cannot do so directly. They have to go through many intermediaries (at the Ministry's information systems department, at the CNAF's managers, etc.). We can make the symmetric assumption. By virtue of the same specialization of administrative knowledge, it is unlikely that CRISTAL's programmers will refer to the letter of the law as we have done and ultimately verify the program's

[37]Interview with DGALN/DHUP/FE4, May 23, 2022.
[38]Interview with DSS/AFPFAT/2B, June 8, 2022.
[39]Interview with DB/D4, July 27, 2022.

compliance with positive law. Rather, they would use specifications and internal memos about the computational content of positive law written elsewhere in the CNAF. Secondly, the procedure for writing standards according to current events described above reinforces the existence of this automation frontier. Indeed, in reaction to current events, ministry agents must quickly write a text and therefore ask the CNAF for their technical constraints, which they record in the writing of the rule itself. They therefore have neither the time nor the technical capacity to carry out counter-expertise and to see the problems of time precision, such as those raised by Section 3.1. As agents from the State budget office told us, "We often hear that this is not possible with the information systems, especially as soon as a new allowance has to be created. We do not have the possibility of making a counter-expertise footnoteInterview with DB/D4, July 27, 2022.". In this automation frontier, the CNAF information systems departments appear to have a great deal of autonomy, since none of the people we met seemed to know why certain modifications in CRISTAL source code appeared to be impossible.

This social division of labor between the writers of the text, the functional specifications and the source code thus constitutes a *communicative circuit between the writing of the code and the law*. This circuit is undoubtedly code-driven in certain respects, but not exclusively: it is also guided by political choices (the reduction in benefits mentioned in Section 1.3) and by many other constraints linked to the bureaucratic writing process.

## 5 LIMITATIONS

After having presented our main results, we wanted to review the limitations of this study and what can or cannot be generalized from French housing benefits to the political effects of programming micro-choices.

First, these results are valid only for a domain of the law whose execution implies a calculation: tax law, social benefits law, and part of labor law. Empirically, these are the domains of the law for which have already been developed large computerized systems dedicated to automatic enforcement. Note that legislation or regulations in these domains directly describe each computational step from the input to the output; they are *process-constrained legal specifications* according to the terminology of Merigoux [2022b]. This study does not generalize to *result-constrained legal specifications* where computerized systems, often based on machine learning technologies, are free to come up with their own way to compute a result, as long as this result satisfies some validity conditions.

Second, the literal programming methodology used to simulate the behavior of the CRISTAL software assumes that the legislation and regulations specifying the software are fully available, and that the programmers can interview, as a last resort, the authors of the legal texts to clarify ambiguities. In our work, we had easy access to the drafters of the French housing law about housing benefits, even though we came from research and not another bureau of the administration. This easy access is explained by the collaborative nature of regulation making for French housing benefits, described in Section 4.2. The access would have been more difficult if we had wanted to work on a part of the law more traditionally shrouded in opaqueness, touching to aspects of the public actions more directly linked to the core of its sovereignty – for example, military pay, another big subject in France [Monin 2018].

Third, the literal programming method is concerned only with the positive law at a given point in time; it is not burdened with previous layers that feed a technical debt and that developers must modify and undo to adapt the existing code to changes in the law. For this simulation, we recognize that we had the privilege to ignore most of the problems encountered by administrative programmers in their daily work. These can only be known through a critical study of the full source

code of the administration, supplemented by interviews with the people involved. Nevertheless, we believe that our new codebase in Catala is somewhat future-proof and could be maintained as French housing benefits law evolved. Changes in the law already happened three times during the course of the publication of this paper, and each time we were able to update our Catala program in a matter of hours.

Fourth and lastly, the literal programming method encounters a limit when the law is expressed with objects that are not just simple text, such as tables or diagrams. Indeed, French housing benefits law contains a lot of tables, and translating them in Catala required to explicit the computational content in them with more lines of code. These highly compact forms of information representation increase the ratio of lines of code to lines of law, in absence of a suitable surface syntax to input them like DMN [OMG Standards Development Organization 2021].

## 6 CONCLUSION: FROM PROGRAMMING CHOICES EXPLICITATION TO RE-SPECIFICATION AND RE-ORIENTATION

Thanks to a methodology mixing critical code studies and sociology, the article addressed the socio-technical issue of unnoticed programming choices within social benefits computation. The general conclusion we draw is that, so far, many scholars have missed the phenomenon of practices, including the ordinary practices of writing, reading and revision of both law and code, in relation to a political, administrative and technological environment. In other words, we bring back here Garfinkel's [1967] "missing what" problem of the study of objective reality of social facts which would encompass the contents of practices. Filling out the "missing what" seems to us indispensable and lead us to specify the three contributions, we want to bring to contemporary debates.

First, the article addresses the "real-world" problem in legal informatics [Governatori et al. 2022; Oskamp and Lauritsen 2002], calling for a re-specification of the subject of research towards more "isomorphism" [Bench-Capon and Coenen 1992]. There is no doubt that in "real-world", we must tackle "poorly-structured knowledge bases" the legislator constantly write – here housing benefits in France. But this is not just a matter of epistemic convenience, it is also a democratic choice: computer scientists have no legitimacy to redefine a better logical order that the rule of law. Nevertheless, the corner-case developed in Section 1.3 illustrates one of the roles that legal informatics can play, in accordance with the rule of law. Moreover, when turning rules into code, we have shown that"bugs" and indeterminate concepts are not the only problems with computation. There is still much to discover in this direction.

Secondly, the article contributes to the philosophy of law, where scholars have focused on the nature of the law and, more precisely, on the alteration that computational technologies inflict to the law [Deakin and Markou 2020]. We encourage pluralizing the conception of the law, and hypothesizing that problems at the bottom of the legal hierarchy may significantly differ from the top. Here, we have considered regulation, even in its most technical and boring aspects, because every adjustment and rules that fuel the computing and distributing of social assistance to millions of beneficiaries is a critical part of the welfare state. In doing so, we have extended the critical stance by identifying that the transition to a computer program is confronted with a problem of precision of the writing (Section 3.1), that the programmers may encounter the temptation to simplify heterogeneous or atypical categories (Section 3.2), and that the corner case reasoning does not match with the administrative process that can be detrimental to users (Section 3.3).

Lastly, the article challenges the sociology of law. Until now, most of the authors that have studied law as a practical activity have focused on Parliament activity, lawyers at work and trials. Our case

study suggests a re-orientation to open the black box of the mundane technologies, to investigate wilder interpretation practices and to deal with computer programs as "enacting law".

## ACKNOWLEDGEMENTS

This paper has benefited from many formal and informal discussions with colleagues. We are in debt to the *Laboratoire Interdisciplinaire Sciences Innovations Sociétés* (LISIS) that has sharpened the sociological and methodological implications of the paper. In Bordeaux, we were lucky enough to be invited by the *Société informatique de France* (SIF) to discuss public algorithmic issues. At the *CRCL 2022: Computational "law" on edge* conference in Brussels, we found a valuable intellectual atmosphere and upliftment. Finally, many thanks to the four reviewers and the CRCL track chair and replier Lyria Bennett Moses for their suggestions and critics.

## REFERENCES

Madeleine Akrich. 1992. The De-scription of Technical Objects. In *Shaping Technology/Building Society. Studies in Sociotechnical Change*, W. Bijker and J. Law (Eds.). MIT Press, 205–224. https://halshs.archives-ouvertes.fr/halshs-00081744

Marie Alauzen. 2019. *Plis et replis de l'État plateforme. Enquête sur la modernisation des services publics en France.* Theses. Université Paris sciences et lettres. https://pastel.archives-ouvertes.fr/tel-02418677

Layman E Allen. 1956. Symbolic logic: A razor-edged tool for drafting and interpreting legal documents. *Yale LJ* 66 (1956), 833.

Association for Computing Machinery. 2020. Artifact Review and Badging Policy. https://www.acm.org/publications/policies/artifact-review-and-badging-current

Benedikt Becker, Jean-Christophe Filliâtre, and Claude Marché. 2020. *Rapport d'avancement sur la vérification formelle des algorithmes de ParcourSup.* Technical Report. Université Paris-Saclay.

Trevor JM Bench-Capon and Frans P Coenen. 1992. Isomorphism and legal knowledge based systems. *Artificial Intelligence and Law* 1, 1 (1992), 65–86.

Xavier Berne. 2018. Les Allocations familiales nous ouvrent le code source de leur calculateur d'aides. *Nextimpact* (2018). https://www.nextinpact.com/article/28136/106298-les-allocations-familales-nous-ouvrent-code-source-leur-calculateur-daides

Pierre Bourdieu. 2005. *The social structures of the economy.* Polity.

Francis Calcoen and Didier Cornuel. 2001. Les aides personnelles au logement en France dans l'histoire des politiques du logement. *Revue des politiques sociales et familiales* 64, 1 (2001), 90–98.

Michel Callon, Bruno Latour, et al. 1981. Unscrewing the big Leviathan: how actors macro-structure reality and how sociologists help them to do so. *Advances in social theory and methodology: Toward an integration of micro-and macro-sociologies* 1 (1981).

Mischa Corsius, Stijn Hoppenbrouwers, Mariette Lokin, Elian Baars, Gertrude Sangers-Van Cappellen, and Ilona Wilmont. 2021. RegelSpraak: a CNL for Executable Tax Rules Specification. In *Proceedings of the Seventh International Workshop on Controlled Natural Language (CNL 2020/21).*

Cour des comptes. 2020. *Rapport public annuel.* https://www.ccomptes.fr/system/files/2020-02/20200225-02-TomeI-aides-personnelles-logement_0.pdf

Simon Deakin and Christopher Markou (Eds.). 2020. *Is Law Computable? : Critical Perspectives on Law and Artificial Intelligence.* Hart Publishing, Oxford.

Laurence Diver. 2021. Interpreting the Rule(s) of Code: Performance, Performativity, and Production. *MIT Computational Law Report* (15 7 2021). https://law.mit.edu/pub/interpretingtherulesofcode https://law.mit.edu/pub/interpretingtherulesofcode.

Vincent Dubois. 2021. *Contrôler les assistés. Genèses et usages d'un mot d'ordre.* Raisons d'agir. 448 pages. https://hal.archives-ouvertes.fr/hal-03462590

Baudouin Dupret and Jean-Noel Ferrie. 2008. Legislating at the shopfloor level: Background knowledge and relevant context of parliamentary debates. *Journal of Pragmatics* 40, 5 (2008), 960–978.

Nel Escher and Nikola Banovic. 2020. Exposing Error in Poverty Management Technology: A Method for Auditing Government Benefits Screening Tools. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW1, Article 064 (May 2020), 20 pages. https://doi.org/10.1145/3392874

Leïla Frouillou, Clément Pin, and Agnès Van Zanten. 2019. Le rôle des instruments dans la sélection des bacheliers dans l'enseignement supérieur. La nouvelle gouvernance des affectations par les algorithmes. *Sociologie* 10, 2 (2019), 209–215.

Harold Garfinkel. 1967. *Studies in ethnomethodology.* Englewood Cliffs, NJ.

Guido Governatori, Trevor Bench-Capon, Bart Verheij, Michał Araszkiewicz, Enrico Francesconi, and Matthias Grabmair. 2022. Thirty years of Artificial Intelligence and Law: the first decade. *Artificial Intelligence and Law* 30, 4 (01 Dec 2022), 481–519. https://doi.org/10.1007/s10506-022-09329-4

Bastien Guerry. 2019. Publication des algorithmes et des codes sources. https://github.com/etalab/algorithmes-publics/blob/master/liste.org

Mireille Hildebrandt. 2020. Code-driven Law: Freezing the Future and Scaling the Past. In *Is Law Computable? : Critical Perspectives on Law and Artificial Intelligence*, Simon Deakin and Christopher Markou (Eds.). Hart Publishing, Oxford, 67–84.

Liane Huttner and Denis Merigoux. 2022. Catala: Moving Towards the Future of Legal Expert Systems. *Artificial Intelligence and Law* (Aug. 2022). https://doi.org/10.1007/s10506-022-09328-5

Ariel Kennan and Sara Soka. 2022. *Benefit Eligibility Rules as Code: Reducing the Gap Between Policy and Service Delivery for the Safety Net.* Technical Report. Beeck Center, Georgetown University. https://beeckcenter.georgetown.edu/report/benefit-eligibility-rules-as-code/

Gilles Kounowski. 2002. L'informatique et le système d'information des Allocations familiales. *Revue des politiques sociales et familiales* 68, 1 (2002), 49–72.

Bruno Latour. 2010. *The making of law: An ethnography of the Conseil d'Etat.* Polity.

Dominique Linhardt and Fabian Muniesa. 2011. Tenir lieu de politique. *Politix* 3 (2011), 7–21.

Michael Lipsky. 1980. *Street-level bureaucracy: Dilemmas of the individual in public service.* Russell Sage Foundation.

Emmanuel Martinais. 2010. L'écriture des règlements par les fonctionnaires du ministère de l'écologie. La fabrique administrative du PPRT. *Politix* 90, 2 (2010), 193–223. https://halshs.archives-ouvertes.fr/halshs-01229937

Denis Merigoux. 2022a. *Observations sur le calcul des aides au logement.* Research Report RR-9485. Inria Paris. 27 pages. https://hal.inria.fr/hal-03781578

Denis Merigoux. 2022b. The Specification Problem of Legal Expert Systems. (Jan. 2022). https://hal.inria.fr/hal-03541637 working paper or preprint.

Denis Merigoux, Nicolas Chataing, and Jonathan Protzenko. 2021a. Catala: A Programming Language for the Law. *Proc. ACM Program. Lang.* 5, ICFP, Article 77 (Aug. 2021), 29 pages. https://doi.org/10.1145/3473582

Denis Merigoux, Raphaël Monat, and Jonathan Protzenko. 2021b. A Modern Compiler for the French Tax Code. In *Proceedings of the 30th ACM SIGPLAN International Conference on Compiler Construction* (Virtual, Republic of Korea) *(CC 2021)*. Association for Computing Machinery, New York, NY, USA, 71–82. https://doi.org/10.1145/3446804.3446850

James Mohun and Alex Roberts. 2020. Cracking the code: Rulemaking for humans and machines. (2020).

Jacques Monin. 2018. Louvois, le logiciel qui a mis l'armée à terre. https://www.franceinter.fr/emissions/secrets-d-info/secrets-d-info-27-janvier-2018

OMG Standards Development Organization. 2021. Decision Model and Notation. https://www.omg.org/dmn/

Anja Oskamp and Marc Lauritsen. 2002. AI in Law Practice? So far, not much. *Artificial Intelligence and Law* 10, 4 (01 Dec 2002), 227–236. https://doi.org/10.1023/A:1025402013007

Edward C Page. 2003. The civil servant as legislator: law making in British administration. *Public administration* 81, 4 (2003), 651–679.

Bruno Palier. 2005. *Gouverner la sécurité sociale: les réformes du système français de protection sociale depuis 1945.* Presses universitaires de France.

François Pellegrini and Sébastien Canevet. 2013. Droit des logiciels. *Presses Universitaires de France* (2013).

Susan Leigh Star. 1990. Power, technology and the phenomenology of conventions: on being allergic to onions. *The Sociological Review* 38, 1_suppl (1990), 26–56.

Susan Leigh Star. 2010. This is Not a Boundary Object: Reflections on the Origin of a Concept. *Science, Technology, & Human Values* 35, 5 (2010), 601–617. https://doi.org/10.1177/0162243910377624 arXiv:https://doi.org/10.1177/0162243910377624

Didier Torny. 2005. L'administration sanitaire entre contraintes techniques et contraintes juridiques. L'exemple des maladies émergentes. *Revue générale de droit médical* 6 (2005). https://halshs.archives-ouvertes.fr/halshs-01249084

Didier Torny. 2011. L'écriture administrative des circulaires : autorégulation professionnelle, produit d'expertise technique ou norme juridique ? In *Droit et régulations des activités économiques : perspectives sociologiques et institutionnalistes*, Bessy, Christian; Delpeuch, Thierry; Pelisse, and Jérôme (Eds.). LGDJ, 255–268. https://halshs.archives-ouvertes.fr/halshs-00719866

Jean-Marc Weller. 1999. *L'Etat au guichet : sociologie cognitive du travail et modernisation administrative des services publics.* Desclée de Brouwer. 254 pages. https://halshs.archives-ouvertes.fr/halshs-00438938

Faïza Zerouala. 2021. La réforme des APL vire au cauchemar pour les allocataires et ses agents. *Mediapart* (2021). https://www.mediapart.fr/journal/france/190621/la-reforme-des-apl-vire-au-cauchemar-pour-les-allocataires-et-ses-agents

## A A CORNER CASE IN THE COMPUTATION OF THE FRENCH HOUSING BENEFITS

The computation formulae for the housing benefits are so complex that the government edits each year a primer about the computation elements, that collects the rules scattered in the various regulations and presents them in a concise and logical order.

First and second, we propose a translation of the excerpts of this primer related to the APL and ALS computations in the specific corner case related in Section 3.3 we want to tackle: an elderly couple (within the meaning of 3° of article D842-16), with one individual earning the minimum wage and the other having no income, which brings their resources to 15,000 € per year. They live in communal housing ("logement-foyer") that meets the criteria for the 1° of D832-25 , in Zone III, and pay a monthly rent of 360 €.

Third and fourth, we highlight how those rules apply by explaining each step of the computation of the APL and ALS on our corner case; The computations for this corner cases have been checked both manually by the authors and automatically by running our Catala program for the housing benefits computation.

In this example, it is preferable for the household to receive the ALF/ALS rather than the APL. Over one year, this represents a difference of about 120 €, a considerable amount for a low-income household. However, since they are eligible to the APL, that is the allowance they will receive from the CNAF, regardless of the real nature of their situation and needs.

### A.1 A Primer on the ALS Computation

The computation of social housing allowance (ALS) is determined according to the formula set out in Article D842-15 of the Construction and Housing Code:

$$\mathrm{ALS} = \mathrm{K} \times (\mathrm{L} + \mathrm{C} - \mathrm{L0})$$

The application of this formula implies the prior determination of several variables.

*The takeover coefficient "K".* Within the framework of the computation of personal housing aid, in the case of communal housing, this computation is based on the formula set out in the 2° paragraph of article D832-25 of the same code which reads as follows:

$$\mathrm{K} = 0.90 - \frac{\mathrm{R}}{\mathrm{cm} \times \mathrm{N}}$$

Where:

- "R" represents the resources of the potential beneficiary.
- "cm" is a multiplying coefficient whose value is set out in Article 30 of the decree of September 27, 2019.
- "N" is the number of units that depends on the number of dependent persons determined according to a scale introduced in d) of 2° of Article D832-25.

K is rounded down to the second decimal.

*The rent equivalence "L".* It depends on the family composition according to the figures set out in Article 43 of the decree of September 27, 2019.

*The lump sum for charges "C".* It is defined by Article 30 of the decree of September 27, 2019.

*The minimum rent "L0".* the computation of which is based on the the minimum equivalence "E0" set out in paragraph 2° of Article D832-26. It is necessary to clarify here the different brackets and rates used in the subsequent computations modes. Indeed, two different sets of brackets and rates are possible in the context of residential accommodation. However, the article detailing the ranges does not mention those used in the second computation mode (the one we are interested in here). The primer about the computation elements of the APL drawn up by the Ministry of Housing in September 2021 nonetheless includes these values, which we have decided to use in our computation. This source does not have full legal value, but it was confirmed to us by an email from the DGLAN/DHUP/FE4 office on May 25, 2022, that these were the correct values.

L0 is determined by the sum of the rates applied to the brackets resulting from the multiplication of the brackets' limits by the number of portions "N" , then increased by the product of a lump sum defined in paragraph II of article 31 of the decree of September 27, 2019 and the number of portions "N", the whole divided by 12. As the wording of this formula is complex, we've included here the corresponding Catala code for reference:

construction_code_regulations.catala_en

```
scope ComputationPersonalizedHousingBenefitCommunalHousing:
  definition brackets_income_d832_26_multiplied equals
    apply for bracket dans brackets_income_d832_26 of
      IncomeBracket {
        -- high: (match bracket.high with pattern
          -- Infinite: Infinite
          -- Income of bracket_high:
            Income content (bracket_high *€ n_number_parts_d832_25))
        -- low: bracket.low *€ n_number_parts_d832_25
        -- rate: bracket.rate
      }
  definition minimal_rent_equivalence equals
    (((sum money for bracket in brackets_income_d832_26_multiplied of
      (if rounded_household_income <=€ bracket.low
      then 0€
      else
        (match bracket.high with pattern
        -- Income of bracket_high:
          (if rounded_household_income >=€ bracket_high
          then
            (bracket_high -€ bracket.low) *€ bracket.rate
          else
            ((rounded_household_income -€ bracket.low) *€ bracket.rate))
        -- Infinite:
          (rounded_household_income -€ bracket.low) *€ bracket.rate)
      ))) +€
    fixed_amount_d832_26 *€ n_number_parts_d832_25) *€ (1,0 /. 12,0)
```

*Final steps.* A lump sum of 5 € should then be deducted from this result and it should be ensured that the minimal expenditure does not need to be subtracted. Then apply the social contribution tax (CRDS[40]) rate, round down to the lowest euro and add the CRDS amount again.

### A.2 A Primer on the APL Computation

The formula used, in the case of communal housing, is set by article D832-24 of the Construction and Housing Code:

$$\mathrm{APL} = \mathrm{K} \times (\mathrm{E} - \mathrm{E0})$$

The application of this formula implies the prior determination of several variables.

*The takeover coefficient "K".* It is defined in Article D832-24 of the Construction and Housing Code:

$$\mathrm{K} = 0.95 - \frac{\mathrm{R} - \mathrm{r} \times \mathrm{N}}{\mathrm{cm} \times \mathrm{N}}$$

Where:

- "R" represents the resources of the potential beneficiary.
- "r" is a coefficient set out in Article 30 of the decree of September 27, 2019.
- "cm" is a multiplying coefficient whose value is set out in Article 30 of the decree of September 27, 2019.
- "N" is the number of units that depends on the number of dependent persons determined according to a scale introduced in e) of 1° of Article D832-25.

K is rounded down to the second decimal.

*The eligible equivalence of rent and rental charges ceiling "E".* It is set by article 27 of the decree of September 27, 2019.

*The computation of the minimum rent and service charge equivalence "E0".* is determined in Article D832-26: it is the sum of rates applied to the brackets resulting from the multiplication of the ranges, listed in article 31, I) of the decree of September 27, 2019, by the number of portions "N", then increased by the product of a lump sum defined in paragraph II of article 31 of the decree of September 27, 2019 and the number of portions "N", the whole divided by 12. It is the same formula as in the computation of the ALS presented in Section A.1, but with a different set of brackets and rates.

*Final steps.* A lump sum of 5 € should then be deducted from this result and it should be ensured that the minimum expenditure does not need to be subtracted. Then apply the CRDS rate, round down to the nearest euro and add the CRDS amount again.

[40]The contribution for the reimbursement of the social debt (CRDS) is a tax created by decree n° 96-50 of January 24, 1996, relative to the reimbursement of the social debt.

### A.3 ALS Computation for Our Corner Case

$$\mathrm{ALS} = \mathrm{K} \times (\mathrm{L} + \mathrm{C} - \mathrm{L0})$$

Where:

- $\mathrm{K} = 0.90 - \frac{\mathrm{R}}{\mathrm{cm} \times \mathrm{N}} = 0.90 - \frac{15000}{1.5 \times 21420.91} = 0,43$
- $\mathrm{L} = 320.73$€
- $\mathrm{C} = 54.22$€

For L0, it is first necessary to multiply the ranges by the number of portions N, which in this case is 1.5, to obtain the brackets, before applying the rates. This yields :

- 0% between 0€ and $1423.03 \times 1.5 = 2134.54$€
- 2.40% between 2134,54€ and $2047.61 \times 1.5 = 3071.42$€
- 20,80% between 3071,42€ and $2629.85 \times 1.5 = 3944.78$€
- 23,30% between 3944,78€ and $4095.05 \times 1,5 = 6142.58$€
- 32,80% between 6142,58€ and 15000€.

Hence,

$$\mathrm{L0} = \frac{3619.29 + 76.32}{12} = 307.97€$$

Therefore:

$$\mathrm{ALS} = 0.43 \times (320.73 + 54.22 - 307.97) = 28.80€$$

After applying the lump sums and the CRDS, we obtain:

$$\mathrm{ALS} = 23.12€$$

### A.4 APL computation for Our Corner Case

$$\mathrm{APL} = \mathrm{K} \times (\mathrm{E} - \mathrm{E0})$$

Where:

- $\mathrm{K} = 0.95 - \frac{\mathrm{R} - \mathrm{r} \times \mathrm{N}}{\mathrm{cm} \times \mathrm{N}} = 0.95 - \frac{15000 - 1217.26 \times 1.8}{13393.40 \times 1.8} = 0,41$
- $\mathrm{E} = 360$€ (the rent is below the rent limit in our corner case)

For E0, it is first necessary to multiply the tiers by the number of portions N, which in this case is 1.8, to obtain the brackets, before applying the percentages. This gives:

- 4% between 0€ and $1948.10 \times 1.8 = 3056.58$€
- 10,40% between 3056,58€ and $2678.71 \times 1.8 = 4821.68$€
- 21,60% between 4821,68€ and $3896.18 \times 1.8 = 7013.12$€
- 26,40% between 7013,12€ and $5357.44 \times 1.8 = 9643.39$€

- 32% between 9643,39€ and $6331.29 \times 1.8 = 11396.32$€
- 48% between 11396,32€ and 15000€

Hence:

$$\text{E0} = \frac{3735.48 + 45.57 \times 1.8}{12} = 318.13€$$

Therefore:

$$\text{APL} = 0.41 \times (360 - 318.13) = 17.17€$$

After applying the lump sums and the CRDS, we obtain :

$$\text{APL} = 12.06€$$