Informatica — Universiteit van Amsterdam

Bachelor Informatica



University of Amsterdam

Machina Pauliana: A neuro-symbolic approach to the actio pauliana in Dutch insolvency law

Daan Boven

30 June 2026

Supervisor(s): David Graus; Gregor Behnke

Signed: Giovanni Sileno

Abstract

The application of legal provisions such as the actio pauliana of Article 42 of the Dutch Bankruptcy Act (Faillissementswet, Fw), which allows an insolvency practitioner to avoid transactions that prejudice creditors, requires both structured rule-based reasoning and sensitivity to open-textured norms developed in case law. Symbolic models capture the structure but cannot account for the open-textured elements, while large language models do the reverse at the cost of interpretability. This thesis investigates how the two can be combined into a neuro-symbolic model of Article 42 Fw.

A symbolic core is written in Catala, a law-as-code language that keeps a one-to-one correspondence between statute and formalisation; the evidentiary presumptions of Articles 43 and 45 Fw are modelled as subscopes, and the open-textured elements, such as prejudice to creditors and the knowledge requirement, are left as boolean inputs that mark where neural components couple to the model. A neural coroutine structures a case's natural-language facts into the model's inputs, and neural subroutines resolve the open-textured elements through case-based reasoning over a corpus of Dutch judgments. That corpus is built from publicly available case law on rechtspraak.nl, segmented by a deterministic finite-state system grounded in Dutch civil procedure, and labelled per legal criterium by a large language model so that the most relevant prior decisions can be retrieved. Because Catala compiles to Python, the symbolic solver calls the neural subroutines directly, returning a verdict together with reasoning grounded in retrieved decisions.

The components are evaluated against manually annotated gold standards. The segmenter attains high precision and recall, and the annotation models perform strongly. The full model, assessed on a small set of eight hand-annotated cases, indicates reasonable performance on most criteria but particular difficulty on the knowledge requirement. Given the sample size these results are indicative rather than conclusive. The central contribution is the integration itself: bringing established legal theory and legal-AI methods together into a single, explainable model of the actio pauliana.

Contents

1 Introduction **7**
- 1.1 Problem . . . 7
- 1.2 Research question . . . 8
- 1.3 Sub-research questions . . . 8

2 Background: legal domain **9**
- 2.1 Article 42 Fw: legal structure and formalisation . . . 9
  - 2.1.1 Formal structure of Article 42 Fw . . . 9
  - 2.1.2 Objectified subjective requirement: knowledge of prejudice . . . 11
  - 2.1.3 Role case law . . . 11
  - 2.1.4 Conclusion . . . 12
- 2.2 Structure of civil judgments . . . 12
- 2.3 Legislative XML . . . 14

3 Background: computational approaches **15**
- 3.1 Symbolic AI . . . 15
  - 3.1.1 Theoretical foundation . . . 15
  - 3.1.2 Modern applications in legal domains . . . 16
  - 3.1.3 Limitations . . . 17
- 3.2 Generative AI . . . 18
  - 3.2.1 Theoretical foundation . . . 18
  - 3.2.2 Applications in legal domains . . . 19
  - 3.2.3 Limitations . . . 21
- 3.3 Neuro-symbolic AI . . . 22
  - 3.3.1 Theoretical foundation . . . 22
  - 3.3.2 Applications in legal domains . . . 24

4 Method **27**
- 4.1 Architecture . . . 27
- 4.2 A symbolic model of the actio pauliana . . . 28
- 4.3 Corpus construction . . . 28
  - 4.3.1 Corpus survey . . . 29
  - 4.3.2 Deterministic segmentation . . . 30
- 4.4 Neuro-symbolic model . . . 32

5 Evaluation **35**

6 Conclusion **39**

7 Discussion **41**

A Systematic coding procedure **43**
- A.1 Segmentation . . . 43
- A.2 RAG pre-retrieval annotation . . . 43

B LLM prompts **45**
B.1 Pre-retrieval annotation 45
B.2 Neuro-symbolic model 46
B.2.1 Coroutine 46
B.2.2 Subroutine: voluntary performance 47
B.2.3 Subroutine: prejudice to creditors 48
B.2.4 Subroutine: knowledge 48

C Cases used for validation **51**

D Catala model of the actio pauliana **53**
D.1 Faillissementspauliana 53
D.1.1 Artikel 42 Faillissementswet 53
D.1.2 Artikel 43 Faillissementswet 55
D.1.3 Artikel 45 Faillissementswet 59

E Catala graph of case execution **61**

CHAPTER 1

Introduction

The field of artificial intelligence and law dates back to at least the 1970s, and most research in the decades since has been devoted to symbolic, rule-based methods, which normalise legal texts into logic-based models that are also intelligible to computers (Rissland, Ashley, and Loui 2003, pp. 6–8) (Ashley 2017, pp. 38–39). Today, increasing attention is devoted to data-driven methods, in which artificial intelligence is used for the automated interpretation of legal texts, document analysis, and the prediction of legal outcomes.

This research focuses on the modelling of the avoidance action in Dutch insolvency law (the *actio pauliana*) as codified in Article 42 of the Dutch Bankruptcy Act (Faillissementswet, Fw). This provision allows an insolvency practitioner to avoid legal acts (transactions) that cause prejudice to creditors and a decrease in the estate. More specifically, if creditors are disadvantaged in their possibilities of recovery as a result of legal transactions entered into by the debtor, such transactions may, under certain conditions, be declared voidable (De Weijs 2010, p. 133). Consider, for example, a sole director who, months before the company is declared bankrupt, sells the company car to himself, not paying the purchase price but setting it off against a loan he claims to have granted the company. The insolvency practitioner may avoid such a transaction: the sale was voluntary, it allowed the director to recover his own claim in full at the expense of the remaining creditors, and, as it took place within one year of the bankruptcy between the company and its own director, knowledge of this prejudice is presumed on both sides.

The legal structure of Article 42 Fw lends itself to formalisation in terms of logical rules. At the same time, the availability of open case law data enables the application of data-driven methods. In this project, these two perspectives are combined by developing a formal logic model of the actio pauliana and integrating it with a data-driven approach. Using publicly available case law from rechtspraak.nl, a dataset can be constructed to support the extraction of relevant features and the training of predictive models.

The aim of this research is therefore to investigate whether a combination of formal legal modelling and data-driven methods can be used to capture and predict the application of Article 42 Fw in judicial practice.

1.1 Problem

The application of legal provisions such as Article 42 Fw (actio pauliana) requires both structured legal reasoning and sensitivity to judicial decisions concerning more open norms. Although the actio pauliana consists of a limited set of formal legal requirements, several of these involve norms that appear to be subjective and are interpreted in judicial practice.

On the one hand, formal logic-based models provide precise and interpretable representations of legal rules, but they are limited in their ability to capture the variability and nuance present in real-world case law. They typically rely on manually specified rules and therefore struggle to account for patterns emerging from previous judicial decisions. As such, they are primarily effective in domains characterised by strict and objective provisions, such as tax law.

On the other hand, data-driven approaches, including machine learning and large language models, are capable of identifying patterns in large collections of legal texts, but they lack the ability to perform explicit rule-based reasoning. As a result, these approaches often suffer from limited interpretability and do not provide the level of legal rigour required in legal analysis.

This creates a fundamental gap: there is currently no approach that simultaneously provides the formal structure required for legal reasoning and the empirical grounding derived from judicial practice. Recent research in neuro-symbolic artificial intelligence seeks to address this limitation by integrating data-driven learning with symbolic reasoning. Neuro-symbolic AI combines the strengths of neural networks, namely their ability to learn from data and recognise patterns, with the strengths of symbolic AI, which provides explicit rule representation and logical inference. As described by Bhuyan et al. (2025), this approach enables systems to both learn from examples and apply formal rules.

However, the application of such hybrid approaches to specific legal domains, such as the actio pauliana in Dutch insolvency law, remains limited. Addressing this gap is important for developing computational models that are both logically grounded and empirically based, and therefore legally meaningful and practically useful.

1.2 Research question

How can formal legal modelling and data-driven methods be combined to model and analyse the application of Article 42 Fw (actio pauliana) in Dutch insolvency law?

1.3 Sub-research questions

To address the aforementioned main research question, the following sub-research questions are formulated:

1. **Symbolic modelling.** How can the legal structure of Article 42 Fw be formally represented in a rule-based, machine-executable model, drawing on *law as code* approaches?

2. **Data pipeline.** How can publicly available case law from rechtspraak.nl concerning Article 42 Fw be retrieved, parsed, and semantically chunked into a structured dataset suitable for applications such as retrieval-augmented generation (RAG)?

3. **Neural subroutine.** How can this dataset be used as a neural subroutine within a symbolic reasoning framework, such that judicial reasoning from prior decisions can be retrieved and applied to support the interpretation of open-textured elements of Article 42 Fw?

4. **Evaluation.** To what extent does the resulting neuro-symbolic model reproduce the legal reasoning of legal practitioners in actio pauliana cases, and where does it diverge from human judicial assessment?

CHAPTER 2

Background: legal domain

This chapter provides the legal background required to support the computational modelling developed in subsequent chapters. It is structured in three parts. Firstly, this chapter focuses on the legal structure of Article 42 Fw, decomposing the actio pauliana into its constitutive elements and identifying which aspects lend themselves to formalisation and which depend on judicial interpretation. Secondly, the conventional structure of Dutch civil judgments under Article 230 Rv is examined, which determines how case law can be segmented for downstream analysis. Lastly, it describes the technical infrastructure through which Rechtspraak data is made available, and the limitations of the existing XML markup that motivate the segmentation approach taken in this research.

2.1 Article 42 Fw: legal structure and formalisation

When a natural person or legal entity is unable to pay its debts to its creditors, it may be declared bankrupt by the courts on request. As long as the (legal) person is not bankrupt, he is in principle free to dispose of his assets. That freedom becomes more constrained as insolvency becomes foreseeable, since such conduct may prejudice creditors (Wessels 2019, pp. 11–12). This creates a grey zone in which the prospect of bankruptcy already casts its shadow over the debtor's actions. Upon bankruptcy, control over the debtor's estate shifts to the insolvency practitioner (curator), who is responsible for managing and distributing the estate among creditors (Pannevis 2026, para. 7.1).

A key instrument available to the insolvency practitioner under the Dutch Bankruptcy Act is the *actio pauliana*. It enables the insolvency practitioner to avoid legal acts (transactions) that cause prejudice to creditors and a decrease in the estate. This is not merely guided by the *pari passu* principle, the equality of all creditors, but primarily by the idea of pool maximisation: assets withdrawn are restored to the estate, enabling the insolvency practitioner to pay more creditors. In this sense, the actio pauliana functions as a mechanism of 'estate reconstruction', ensuring that the pool of assets available for distribution reflects the situation that would have existed in the absence of the disputed legal act(s) (De Weijs 2010, p. 133).

From a computational perspective, the actio pauliana can be understood as a rule-based system with defined conditions, such as the existence of a legal act, its voluntary nature, and the presence of creditor prejudice. However, some elements require interpretation and have been developed in case law. This combination of formal structure and interpretative flexibility makes the doctrine particularly suitable for a hybrid modelling approach that combines symbolic reasoning with data-driven methods.

2.1.1 Formal structure of Article 42 Fw

The transaction avoidance specific to Dutch bankruptcy proceedings is incorporated in the Dutch Bankruptcy Act ("Faillissementswet", *Fw* in short) under Articles 42-51. This research focuses on Article 42 Fw, which concerns voluntary legal acts. Article 47 Fw, by contrast, governs legal

acts performed under a pre-existing obligation. Since such acts are in principle required to be performed, their avoidance is subject to stricter conditions. For reasons of legal certainty, the legislator has made it more difficult to challenge such obligatory acts (Vriesendorp 2024, p. 244). Accordingly, this research limits itself to Article 42 Fw, as it provides the most suitable basis for computational modelling.

Article 42 Fw can be decomposed into different elements (Vriesendorp 2024, p. 245):

- The transaction is a legal act.
- The transaction was voluntary.
- It has caused prejudice to creditors.
- The debtor knew or should have known that the prejudice to creditors resulted from the legal act.
- When the transaction was for consideration, the counterparty of the transaction knew or should have known that it would lead to prejudice to creditors.

Articles 43 and 45 Fw introduce evidentiary presumptions that assist the insolvency practitioner in establishing the knowledge requirement. These presumptions apply in specific situations, such as transactions at an undervalue or transactions with affiliated parties, and will be discussed further below.

Combining the elements outlined above into a schematic flow chart results in the figure below.

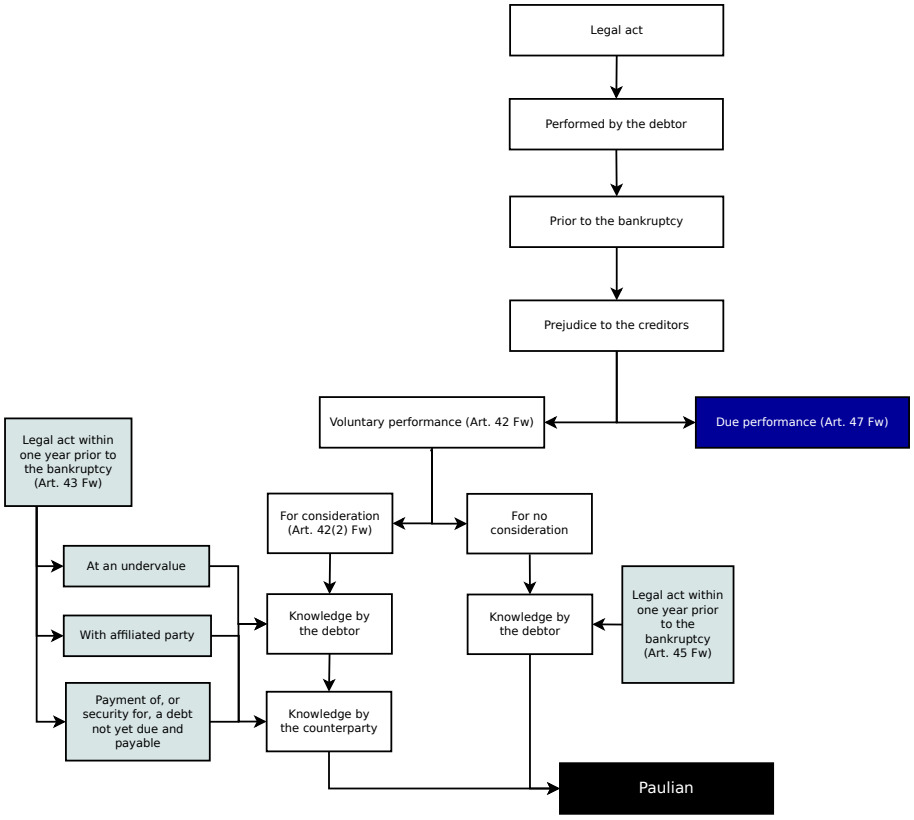


Figure 2.1: Actio pauliana schematic model based on Pannevis (2026, para. 7.2.1) and translation based on Warendorf and Thomas (2015).

2.1.2 Objectified subjective requirement: knowledge of prejudice

The actio pauliana can only be successful when there is knowledge of prejudice to creditors on the part of the debtor, and, in the case of a legal act for consideration, also on the part of the counterparty (Pannevis 2026, para. 7.2.2.6). At first sight, this appears to be a subjective criterion. However, the knowledge criterion has been objectified by equating actual knowledge with an 'ought to have known' standard, meaning that the insolvency practitioner does not need to prove the actual inner state of mind of the debtor or counterparty (Vriesendorp 2024, p. 249).

According to the Supreme Court, the required knowledge is present when, at the time of the transaction, bankruptcy and a deficit in the estate were foreseeable with a reasonable degree of probability (Pannevis 2026, para. 7.2.2.6). At the same time, this does not make the requirement fully objective. The assessment remains tied to the moment at which the transaction was concluded, and concerns what the parties knew or ought to have realised in light of the circumstances at that time. In that sense, the literature describes this as an objectified subjective requirement. Vriesendorp further argues that, as a lower bound, the probability of bankruptcy should not be less than 50%, although this should be understood as a doctrinal interpretation rather than as a fixed judicial threshold (Vriesendorp 2024, pp. 249–250).

To assist the insolvency practitioner in establishing the knowledge requirement, Articles 43 and 45 Fw introduce rebuttable evidentiary presumptions. These operate in specific situations, such as transactions at an undervalue, transactions with affiliated parties, and the payment of, or provision of security for, a debt not yet due and payable. They are subject to the condition that the legal act occurred within one year prior to bankruptcy (Vriesendorp 2024, p. 251). While these presumptions do not remove the requirement entirely, they render its application more rule-like and therefore more feasible for computational modelling.

2.1.3 Role case law

Although Article 42 Fw can be decomposed into a set of seemingly clear conditions, several of its core elements are dependent on interpretation developed in case law. In particular, the following requirements are not exhaustively defined in the statute and have been shaped to a large extent by jurisprudence:

- prejudice to creditors (*benadeling*);
- knowledge of such prejudice by the debtor (and, where applicable, the counterparty).

Prejudice to creditors

A central concept within Article 42 Fw is that of prejudice (*benadeling*). Case law has clarified that this requirement should not be assessed at the moment the legal act is performed, but at the moment the actio pauliana is invoked. This is known as the doctrine of indirect prejudice (*middellijke benadeling*). According to established jurisprudence, prejudice must be present at the time when the insolvency practitioner exercises the avoidance action, meaning that at that moment the possibilities of recovery for creditors are less favourable than they would have been in the absence of the legal act (Wessels 2019, p. 66).

To determine whether such prejudice exists, case law applies a hypothetical comparison. The actual situation, in which the contested transaction has taken place, is compared with a hypothetical situation in which the transaction did not occur. Importantly, this comparison does not involve speculation about all possible alternative transactions, but rather an assessment of what would most likely have happened if the legal act had not been performed (Vriesendorp 2024, p. 247). Judicial decisions illustrate how this comparison is applied in practice across different types of transactions (Wessels 2019, pp. 70–73).

Furthermore, it is not required that all creditors are prejudiced. It is sufficient that the transaction adversely affects the recovery possibilities of at least one creditor (Wessels 2019, p. 68).

Case law also distinguishes between two main forms of prejudice. First, prejudice may consist of a reduction of the estate, meaning that fewer assets are available for distribution among creditors. Second, prejudice may arise from a disruption of the ranking of creditors, for example

where one creditor is placed in a more favourable position at the expense of others (Wessels 2019, p. 76).

Knowledge of prejudice

The requirement of knowledge of prejudice is likewise shaped by case law. Although Article 42 Fw refers to knowledge, this is interpreted in an objectified manner, as discussed above. The standard applied is not limited to actual knowledge, but includes situations in which the debtor or counterparty *ought to have known* (*behoren te weten*) that the legal act would result in prejudice to creditors (Wessels 2019, p. 88).

As discussed above, the Supreme Court has specified this requirement through a foreseeability test, according to which knowledge is present where bankruptcy and a deficit were foreseeable with a reasonable degree of probability. This standard has been consistently applied in subsequent case law and provides a more concrete interpretation of the otherwise open-textured knowledge requirement (Pannevis 2026, para. 7.2.2.6) (Wessels 2019, pp. 93–94).

2.1.4 Conclusion

The information of Article 42 Fw outlined above provides a meaningful basis for computational modelling of the Dutch actio pauliana. Its statutory structure can be decomposed into a set of identifiable legal conditions. These elements can be represented in a schematic model and therefore lend themselves, at least in part, to formal and logical modelling.

At the same time, the analysis also shows that the practical application of Article 42 Fw cannot be reduced to statutory text alone. Core elements such as prejudice (*benadeling*) and knowledge of prejudice have been substantially developed in case law.

This has important implications for modelling. Some aspects of Article 42 Fw are sufficiently structured to be formalised in a rule-based system, including the ordering of conditions and the role of evidentiary presumptions. Other aspects are more open-textured and fact-sensitive, and are therefore less suitable for purely symbolic representation. For those elements, case law is not just illustrative, but necessary to understand how the rule operates in practice.

2.2 Structure of civil judgments

Dutch civil judgments are generally drafted according to a fixed structure. Although this structure in practice varies between judicial instances (district court, court of appeal, Supreme Court) and even between courts of the same instance, a general structure of the civil judgment can be derived from Dutch civil procedural law and the corresponding literature. At first instance, before the district court, this structure originates in Article 230 of the Dutch Code of Civil Procedure ("Wetboek van Burgerlijke Rechtsvordering", *Rv* or *CCP* in short), which reads:

> **Article 230**
>
> 1. The judgment states:
>   a. the names and the domicile of the parties, and the names of their authorised representatives or lawyers;
>   b. the steps taken in the action;
>   c. the conclusion in the writ of summons and the parties' statements;
>   d. the conclusion of the Public Prosecutor's opinion, where the Public Prosecutor has been heard;
>   e. the reasons for the decision, including the facts on which the decision is based;
>   f. the decision;
>   g. the name of the judge, or in the event of a chamber consisting of more than one judge, the names of the judges who issued the judgment;
>   h. the date of the judgment.

(Translation from Oranje, Frakes, and Visser 2025)

According to the literature, Article 230 Rv is applied by opening the judgment with the case number and the parties. This is followed by the course of proceedings, an account of what has happened in the proceedings up to that point. The judgment then sets out the claim and the parties' statements, often referred to simply as the dispute. The court subsequently lists the established facts, being those asserted by one party and either admitted or insufficiently disputed by the other. The court then turns to the grounds of the decision, containing its reasoning, and closes with the decision (also called the *dictum*, or operative part), often under the heading *Beslissing* or introduced by the phrase "recht doende ..." (Gras, Hendrikse, and Jongbloed 2024, sec. 9.3.1) (Snijders et al. 2022, p. 210).

Ahsmann (2020) examines in more detail how judges draft these parts in practice. Under the heading *Verloop van het geding* or *Verloop van de procedure*, the judge gives a concise, professional account of all procedural steps and their dates (Ahsmann 2020, p. 229). The relevant facts are included under the title *De vaststaande feiten* or simply *De feiten*. This part should contain no legal matters, such as claims or legal findings. The claim, its basis, and the defence are usually set out under the heading *Het geschil* or *De vordering*, depending on the judgment model used. The judge in any event states the claimant's claim, and sometimes also includes the defendant's defence (Ahsmann 2020, pp. 241–248). The judge's reasoned assessment of the claim follows under *De beoordeling*. Unlike in criminal or administrative law, no fixed decision scheme is used here (Ahsmann 2020, p. 251).

Through the following linking provision of Article 353 Rv, the court of appeal is likewise bound to follow Article 230 Rv when drafting its judgment:

Article 353

1. Except where this Title or the law provides otherwise, Title 2 applies *mutatis mutandis* to appeals. However, the parties must be represented by a lawyer, Article 131 does not apply, and no counterclaim may be initiated.

(Translation from Oranje, Frakes, and Visser 2025)

The civil divisions of the courts of appeal maintain a professional standard for the drafting of judgments. Under this standard, an appellate judgment opens with a short introduction to what the case is about, usually under the heading *De zaak in het kort*. The course of proceedings follows as an itemised list of the documents filed and the hearings held. The judgment then presents the relevant facts, as established by the court of appeal and insofar as relevant to the decision, followed by the claims, the decision of the district court, and the dispute on appeal; the district court's considerations and its decision need only be summarised briefly. Finally, the judgment turns to the grounds of the decision, containing the court of appeal's legal assessment and the operative part (Bartelds et al. 2020, pp. 10–16). The sections set out above for the district court thus reappear at the court of appeal, albeit possibly under different wording. Additional elements may moreover appear, given that the court of appeal refers to what was decided at first instance.

In cassation before the Supreme Court, judgments likewise follow a fixed structure, one that shows clear similarities with the foregoing. Since 2019, the Supreme Court uses the following structure for its judgments. First, it opens with a formal part stating the date, the chamber, the parties, and the like. This is followed by the course of proceedings, comprising a reference to the relevant decisions in the fact-finding instances (district court, court of appeal) and a concise account of the proceedings in cassation. An overview of the points of departure and the facts follows, including the relevant parts of the contested decision, usually under the heading *Uitgangspunten en feiten*. The Supreme Court then sets out the actual reasoning behind its decision (*Beoordeling van het middel*), and the judgment closes with the operative part under the heading *Beslissing* (De Groot and Van der Voort Maarschalk 2026, no. 314). Until 2019, the course of proceedings was instead split into *Het geding in feitelijke instanties* and *Het geding in cassatie*, after which the actual reasoning opened by recalling the relevant factual points of departure. These facts have already been established by the courts below, as the Supreme Court

does not establish facts itself; that occurs exclusively at first instance and on appeal (De Groot and Van der Voort Maarschalk 2019, no. 314) (Asser 2018, pp. 116–117).

In conclusion, the following substantive parts recur across all judicial instances, leaving aside the formal opening with the parties and similar details, albeit at times under different headings and not necessarily all present as separate sections:

- *Procesverloop* — course of proceedings;
- *Feiten* — established facts;
- *Geschil* — dispute;
- *Overwegingen* — grounds of the decision;
- *Beslissing* — decision.

This five-part structure serves as the point of departure for the further processing of judicial decisions on the actio pauliana in this thesis, with the recognition that it is not exhaustive and leaves aside some of the legal nuances discussed above.

2.3 Legislative XML

The infrastructure that enables computational analysis of Rechtspraak data has developed in stages. Opijnen (2014, pp. 235–236) records that from 2004 Rechtspraak.nl delivered all anonymised judicial decisions deemed legally or socially relevant as XML free of charge, and that since June 2013 the ECLI-register and the published judgments have been available as open data with technical documentation. Each decision is identified by its European Case Law Identifier (ECLI).

The richness of the markup has long lagged behind its accessibility. Trompper and Winkels (2016) reported that 78.7% of judgment texts carried no `section` tag at all, with substantive structural markup becoming common only from around 2013. The current specification (IVO Rechtspraak 2019, pp. 7–17) retrieves documents through a RESTful web service, ECLI-metadata via the `zoeken` endpoint, individual judgments via the `content` endpoint, returning XML under an `open-rechtspraak` root that holds Dublin Core metadata and, where available, the judgment text under a `uitspraak` or `conclusie` element. The text schema derives from DocBook 5.0, and `section` elements may carry a `role` attribute identifying their function within the judgment.

Even so, the data are not clean. Tjong Tjin Tai (2021, pp. 166–170) describes a Tilburg Law School pipeline of collection, integration, scrubbing and transformation, observing that documents contain typographical errors, that the `uitspraak` element is not uniform across years, and that internal delimiters such as the heading of the dictum appear under many surface forms, *De beslissing*, *rechtdoende*, and others. Of the `role` attribute, moreover, only three values are systematically populated across the corpus: *procesverloop*, *overwegingen* and *beslissing*. As the structure of a Dutch civil judgment shows (Section 2.2), a complete *vonnis* contains more than these three parts, in particular a separate *feiten* section and the parties' *conclusies*, neither of which carries a role. Downstream analysis that depends on locating these sections therefore cannot rely on the markup alone, and must fall back on some form of segmentation (Section 4.3.2).

CHAPTER 3

Background: computational approaches

This chapter reviews the computational approaches that inform the modelling developed in this research. It is structured in three parts. Firstly, it examines symbolic AI: the rule-based and logic-based methods that have long dominated the field of AI and law. Secondly, it turns to large language models and generative AI, which approach legal text through data-driven learning. Lastly, it discusses neuro-symbolic AI, which seeks to combine symbolic reasoning with neural learning and which underpins the hybrid approach taken in this thesis.

3.1 Symbolic AI

3.1.1 Theoretical foundation

The field of AI and law dates back at least to the 1970s: the 1970 *Stanford Law Review* article 'Some Speculation about Artificial Intelligence and Legal Reasoning' discussed modelling legal reasoning for legal analysis and argumentation (Rissland, Ashley, and Loui 2003, p. 6). Most research over the past five decades has been devoted to symbolic AI and law, that is, to symbolic, rule-based methods. Modelling statutory and case-based reasoning consists of normalising legal texts into a logic-based model that is also intelligible to computers (Ashley 2017, pp. 38–39, 43).

The British Nationality Act experiment. A landmark experiment in this field is the work of Sergot et al. (1986) at Imperial College London, which translated a large part of the British Nationality Act 1981 (BNA) into a Prolog (logic) program. The formalisation relied on Horn clauses, a subset of first-order logic (Sergot et al. 1986, p. 370). The authors extended these Horn clauses to handle negation, going as far as negation as failure (Sergot et al. 1986, pp. 379–382). Their work also showed that, rather than being written to strict program specifications, such programs are developed through trial and error (Sergot et al. 1986, p. 376). The experiment moreover proved to be a rich source of problems for logic-based knowledge representation (Sergot et al. 1986, p. 379) (Rissland, Ashley, and Loui 2003, p. 7) (Ashley 2017, pp. 48–49).

From propositional to predicate logic. As the BNA modelling shows, the normalisation of legal texts can begin by identifying atomic propositions. Allen and Engholm (1978) noted that these propositions should be replaced with labels (S_1, S_2, and so on), after which propositional logic can be used to formulate the syntax of the statute; the text of the atomic, substantive propositions is then restored to validate the logic model. A version normalised in propositional logic can be visualised as a flow chart, which makes it easier to understand and allows it to be executed on a computer (Ashley 2017, pp. 41–43). Predicate logic can additionally be introduced, as the BNA experiment illustrates, to express both content and syntax in a form intelligible to computer-based systems. Horn-clause logic, a subset of first-order logic, allows such legal statements to be expressed. Consider H.L.A. Hart's example 'vehicles are not permitted in the park', expressed as a Horn clause in Prolog (Ashley 2017, pp. 44–45):

$$violation(X, S) \text{ :- } vehicle(X),\ park(S),\ in(X, S).$$

Catala. Beyond Prolog, computational law has produced new languages that translate law into an executable specification. Catala is designed to follow the existing structure of legal texts, 'enabling a one-to-one correspondence between a legal paragraph and its corresponding translation in Catala' (Merigoux, Chataing, and Protzenko 2021, p. 3). Logic forms its core system, and Catala compiles to a generic lambda-calculus, which allows it to be translated into any existing language (Merigoux, Chataing, and Protzenko 2021, p. 3). Without covering the full language definition, Catala differs from most functional languages in two respects. First, it allows context variables to be defined with an optional default case and a number of specialised cases, which may be prioritised or non-overlapping; this is known as prioritised default calculus. This representation was simplified from the original prioritised default logic, which made rules 'overly complicated' (Merigoux, Chataing, and Protzenko 2021, p. 18). Second, Catala is entirely declarative, owing to the out-of-order nature of its definitions: the compiler computes a suitable dependency order for the definitions in a program. This design choice was feasible because law does not involve general recursion (Merigoux, Chataing, and Protzenko 2021, p. 11).

Rule-based and case-based reasoning. The first initiatives in computational law were geared towards rule-based reasoning (RBR) systems. The BNA modelling is again illustrative: the Act laid down by the legislator contains rules, for which a reasoning system was developed. Case-based reasoning (CBR), by contrast, concerns case law and reasoning by analogy (Rissland, Ashley, and Loui 2003, p. 8). Modern approaches frequently combine the two in a hybrid CBR–RBR design, acknowledging that meaningful legal reasoning often involves applying both legal statutes and previous judicial decisions (Rissland, Ashley, and Loui 2003, p. 11). The balance between the two also depends on the legal system, that is, on whether it is a civil-law or common-law jurisdiction.

Information extraction. Case-based reasoning systems rely on accurate information extraction from case law. Within the symbolic paradigm, a judicial decision can be dissected into relevant sections using cascaded finite-state transducers, which chain multiple finite-state automata together so that the output of one becomes the input of the next. FASTUS applies this concept to extract information from free text. It comprises four stages: a keyword scan (triggering), the recognition of phrases, the recognition of patterns of interest, and finally the merging of incident structures across the text (Hobbs et al. 1993, p. 133). Its advantages are that it is conceptually simple, fast at runtime, and effective precisely because of its cascading structure (Hobbs et al. 1993, p. 137). Jurafsky and Martin (2026a, p. 19) describe this as a rule-based approach to information extraction, in which cascades or regular expressions recognise chunks produced by previous stages on the basis of parts of speech, trigger words, or classes.

3.1.2 Modern applications in legal domains

The law-as-code paradigm and rule-based reasoning. Recent initiatives in the Netherlands, such as RegelRecht, led by the Dutch Ministry of the Interior and Kingdom Relations, build on the observation as noted by Ashley (2017, p. 42) that machine-executable, normalised versions of laws can aid understanding of those laws. They can moreover improve the transparency, consistency, and traceability of the application of law, reducing interpretative discrepancies between legal text and implementation (Ministerie van Binnenlandse Zaken en Koninkrijksrelaties 2025a). Dutch efforts in this area, particularly the use of RBR systems by local and central government agencies, can be traced back to the 1990s (Rissland, Ashley, and Loui 2003, p. 7). Another field that lends itself to this paradigm is tax law, which contains fewer open-textured terms and is largely 'calculative'. In the Regelspraak project, Dutch tax law has been rendered in a controlled natural language intelligible to both people and computers (Ministerie van Binnenlandse Zaken en Koninkrijksrelaties 2025b). None of these projects, however, addresses the modelling of Dutch civil law, and Dutch insolvency law in particular.

Catala is used primarily for tax law and has been tested on the French tax code, having been developed at the French national research institute Inria. Its main practical advantage is a systematic approach in which literate programming (annotating the law line by line) and pair programming (a computer scientist and a legal scholar working side by side) are the norm *ex ante* (Huttner and Merigoux 2022, §4.2.1) (Merigoux, Chataing, and Protzenko 2021, p. 23). This helps address a concern raised by Ashley (2017, p. 46): who has the authority to choose among the possible normalised versions of a statute? Catala answers this collaboratively, by making the formalisation readable and usable for computer scientists and legal scholars alike. In practice, Catala compiles to many programming languages, allowing execution on both modern and legacy systems without requiring an API (Huttner and Merigoux 2022, §5.2.1–5.2.2) (Merigoux, Chataing, and Protzenko 2021, p. 22).

Information extraction from case law. Jackson et al. (2003) developed a symbolic approach for extracting rulings from court opinions and retrieving relevant prior cases, building on a cascade of transducers to structure the information (Jackson et al. 2003, pp. 239–240). Although focused on common-law systems, their work recognised that the textual cues marking a section vary from court to court, which complicates extraction (Jackson et al. 2003, p. 245). They adopted an approach with multiple levels of processing, including syntactic and semantic processing (Jackson et al. 2003, p. 249), and used a variant of the CYK parsing algorithm with the power of a pushdown automaton, tolerating both lexical and structural ambiguity (Jackson et al. 2003, p. 251). The final system could parse a large corpus of past cases, but it could not extract information merely implied by the text, as humans do. This stemmed from a lack of inference, which prevented the system from drawing the simple conclusions a writer expects readers to make (Jackson et al. 2003, pp. 286–288). Modern approaches with inferential capabilities, such as Large Language Models, may be able to address this in part.

3.1.3 Limitations

Difficulties of the BNA program. Sergot et al. (1986) themselves noted limitations of their symbolic approach to modelling the British Nationality Act 1981. Some stemmed from the use of Prolog and extended Horn clauses, such as the inability to handle disjunctive or negative conclusions (Sergot et al. 1986, p. 379). Extended Horn-clause logic also copes poorly with counterfactual conditionals, which can occur frequently (Sergot et al. 1986, p. 382). Furthermore, previously formulated rules or predicates sometimes had to be reformulated, and open-textured terms had to be addressed (Ashley 2017, p. 49). Their solution to this last problem was straightforward: the system prompts the user to enter True or False for the open-textured term in the inquiry at hand (Ashley 2017, p. 51). Given advances such as Large Language Models and Retrieval-Augmented Generation, such a solution would today be considered rudimentary.

Vagueness and ambiguity. Symbolic approaches to computational law, particularly the earlier experiments, struggle with vagueness and with semantic and syntactic ambiguity. Vagueness is a semantic uncertainty about 'precisely where the boundary is with respect to what a term does and does not refer to' (Allen and Engholm 1978, p. 382) (Ashley 2017, p. 39). Semantic ambiguity is 'uncertainty between relatively few distinct alternatives' (Allen and Engholm 1978, p. 383) (Ashley 2017, p. 39). Syntactic ambiguity, finally, arises because legal statutes do not always follow a single, coherent logical structure: natural-language text cannot explicitly specify the scope of logical connectives such as *if*, *and*, *or*, and *unless*. When deciding how to model a statute symbolically, Allen stressed that a systematic procedure for generating normalised versions of statutes is crucial, not least to familiarise legislators and legal scholars with that procedure (Ashley 2017, p. 46). Modern approaches such as the Catala language aim to address this by enabling pair programming between legal scholars and computer scientists (Huttner and Merigoux 2022, §4.2.1).

When the rules run out. Anne Gardner observed in 1984 that open-textured elements, and the relationship between technical and common-sense concepts, can lead to the problem of 'the rules running out': the antecedent of a rule invokes a predicate that no further rule defines

(Rissland, Ashley, and Loui 2003, p. 7). This core limitation of symbolic reasoning motivates the introduction of a neural component, which could enable reasoning with the open-textured nature of legal concepts and inference based on 'common sense' concepts. The knowledge requirement on the part of the debtor in the Dutch transaction avoidance is precisely an instance of the problem Gardner identified: the interpretation of such criteria is left to judges and requires more than rule-based reasoning alone.

3.2 Generative AI

3.2.1 Theoretical foundation

Large Language Models

Pre-training and the three architectures. Large language models rest on a central idea: *pre-training*, the acquisition of knowledge about natural language by iteratively predicting tokens in large amounts of text (Jurafsky and Martin 2026c, pp. 1–4). Three model architectures are commonly distinguished; this section introduces all three, although the remainder of the thesis builds on only two. The *decoder* takes a sequence of tokens as input and generates one output token at a time, feeding each new token back into the input. It powers models such as GPT, Claude, and Mistral (Jurafsky and Martin 2026c, pp. 4–5). The *encoder* also takes a sequence of tokens, but outputs a vector representation for each one. Models in the BERT family are encoders: they do not generate text, but are typically used to build classifiers and embeddings, as discussed below in the section on retrieval-augmented generation. The *encoder–decoder*, finally, maps an input sequence of tokens to an output sequence, and can therefore translate between different kinds of tokens, for instance between text and speech (Jurafsky and Martin 2026c, pp. 4–5). As this research concerns natural-language text, the encoder–decoder architecture is not considered further.

Prompting and in-context learning. A base language model becomes considerably more useful once it is trained to follow instructions. Continuing the training of a base model on a set of instructions and corresponding answers is known as *instruction tuning*. When a user queries such an instruction-tuned model, finding the most effective prompt to elicit a useful response is known as *prompt engineering*. Performance can be improved by including annotated examples in the prompt that demonstrate how to solve the task, known as *few-shot prompting*, whereas omitting such demonstrations is *zero-shot prompting* (Jurafsky and Martin 2026c, pp. 6–7). A model can in this way learn during prompting rather than during pre-training, without updating its underlying parameters; this is known as *in-context learning*. Models also typically receive a *system prompt*, the first instruction they are given, which sets the role of the model (Jurafsky and Martin 2026c, pp. 8–9).

Sampling parameters. Because a large language model samples its next token, several parameters govern that sampling process. *Temperature* sampling reshapes the model's output distribution: a lower temperature sharpens it, making likely tokens more dominant and unlikely ones rarer, and so yields more deterministic output, whereas a higher temperature flattens it (Jurafsky and Martin 2026c, p. 12). *Top-k* sampling instead restricts the choice to the k most likely tokens, renormalises their probabilities, and samples from that reduced pool (Jurafsky and Martin 2026e, p. 19). *Top-p* sampling truncates the distribution by probability mass rather than by count, keeping however many tokens are needed to reach a target share p; tying the cutoff to probability lets the candidate pool adapt to context, widening when the model is uncertain and narrowing when one token clearly dominates (Jurafsky and Martin 2026e, p. 20).

Retrieval-Augmented Generation

Information retrieval. Plain prompting, as described above, has well-known limitations, including hallucination, poor calibration, and static knowledge; these are discussed further in Section 3.2.3. One way to supply a model with specific, current knowledge is *ad hoc retrieval*: selecting relevant

documents from a collection, for example previous judicial decisions, so that they can be placed in the prompt. In a simple approach, each document is represented as a vector of the word counts it contains, and documents are ranked against the query by cosine similarity (Jurafsky and Martin 2026b, p. 7). More refined sparse methods weight each term using schemes such as tf-idf and BM25 (Jurafsky and Martin 2026b, pp. 7–9). Their shared weakness is that they match only on exact word overlap between query and document, the *vocabulary mismatch problem* (Jurafsky and Martin 2026b, p. 13). Dense retrieval addresses this by embedding text with encoders such as BERT. A full *cross-encoder* encodes every query–document pair jointly:

$$\mathbf{z} = \text{BERT}(\mathbf{q}; [\text{SEP}]; \mathbf{d})[\text{CLS}]$$

$$\text{score}(q, d) = \text{softmax}(\mathbf{U}(\mathbf{z}))$$

This is computationally expensive, as every document must be re-encoded for each query. A *bi-encoder* instead pre-computes the document embeddings in advance and encodes only the query at query time; this is less accurate, but considerably cheaper (Jurafsky and Martin 2026b, p. 13).

Naive RAG. *Retrieval-augmented generation* (RAG) builds on this retrieval machinery: documents retrieved from a collection are inserted into the prompt to ground and improve the model's output (Jurafsky and Martin 2026b, p. 3). Lewis et al. (2021) laid the foundation for the approach, combining the *parametric* memory of a pre-trained generative model with the *non-parametric* memory of a retrievable document index, coupled through a general-purpose fine-tuning procedure (Lewis et al. 2021, p. 2). They proposed two variants: the RAG-sequence model conditions an entire generated answer on a single retrieved document, whereas the RAG-token model may draw on a different document for each token, so that one answer can combine content from several documents (Lewis et al. 2021, p. 3). Gao et al. (2024, p. 3) later termed this early approach *Naive RAG*: a standard pipeline of indexing, retrieval, and generation.

Advanced RAG. Naive RAG has several weaknesses. Retrieval may return passages of poor precision or recall, and the model may over-rely on the retrieved text, echoing it rather than integrating it with the task at hand (Gao et al. 2024, p. 3). *Advanced RAG* addresses these issues by adding pre-retrieval and post-retrieval stages (Gao et al. 2024, p. 3). In the pre-retrieval stage, indexing is improved, for example by attaching metadata on which document selection can be based; attaching metadata about the legal criteria a judicial decision concerns is a particularly relevant strategy in the context of this thesis. In the post-retrieval stage, the retrieved material is integrated with the query as effectively as possible, through techniques such as reranking the retrieved passages and compressing their context (Gao et al. 2024, p. 4).

Chunking strategies. RAG is typically applied to unstructured, textual data, which is also prevalent in the legal domain, for instance in case law. During indexing, this unstructured text is split into *chunks*, each of which is passed through an encoder and stored as a vector embedding in a database. The most common strategy, *fixed-size chunking*, splits the text every fixed number of tokens. In the legal domain this is often undesirable, since judicial decisions are organised into logical sections that serve distinct purposes and cannot simply be split mid-section or mid-sentence. *Semantic chunking* instead splits the text along its meaningful, logical sections (Gao et al. 2024, pp. 7–8). This strategy aligns well with the structure of judicial decisions set out in Section 2.2.

Embeddings and encoders. The choice of embedding model matters as well. Specialised domains such as law contain a substantial amount of proprietary jargon, and therefore benefit from a fine-tuned embedding model, or encoder, rather than a general-purpose one (Gao et al. 2024, p. 9). LEGAL-BERT is one example of such a domain-adapted encoder.

3.2.2 Applications in legal domains

The applications surveyed below are ordered along a spectrum from using large language models to structure legal text toward using them to generate grounded legal output: from text annota-

tion, through structured element extraction, to retrieval-augmented question answering and the drafting of legal documents.

A rigorous process for LLM-based text annotation. Törnberg (2024) developed a systematic procedure for data annotation using large language models. Although the approach is geared towards the social sciences (Törnberg 2024, p. 2), it transfers to legal text annotation, since both fields work with unstructured textual data whose coding requires interpretative judgement. LLMs are non-deterministic, prone to bias, and may yield unreliable results, but, as Törnberg argues, this is not a reason to reject their use in annotation altogether; rather, it is a reason to embed them in a rigorous process that manages those weaknesses. He identifies at least six factors relevant to choosing a model (Törnberg 2024, pp. 2–4). The procedure itself runs iteratively: the concepts to be annotated are defined, a small training set is coded (or a pre-coded one is reused), disagreement between coders and between coder and model is examined, and the codebook and prompt are refined until the output is ready for validation (Törnberg 2024, pp. 7–8). Validation comes only after the prompt is finalised, since using validation to revise the prompt can produce inflated results (Törnberg 2024, p. 9). The prompt itself should include an explicit uncertainty option and require JSON-formatted output, since JSON is straightforward for an LLM to produce and easy to parse downstream (Törnberg 2024, p. 11).

Combining rule-based parsing with LLM extraction. As established in Section 3.1.2, combining symbolic parsing with LLM-based information extraction can be a powerful approach to structuring data. Li et al. (2025) proposed LEE, a three-stage scheme for extracting legal elements from legal documents: document segmentation, element extraction, and aggregation (Li et al. 2025, pp. 1372–1373). Their approach combines rule-based parsing for fixed-pattern fields, a combination of regular expressions, keyword matching, and syntactic analysis, with context-prompting learning (CPL) for semantically complex fields (Li et al. 2025, pp. 1373–1374). The rule-based component is a descendant of earlier symbolic pipelines such as FASTUS (Hobbs et al. 1993). The CPL component is itself specified through a clear goal, contextual guidance on where to find the relevant information, domain-specific rules, and a JSON output specification (Li et al. 2025, p. 1375).

Case-based reasoning and RAG. The retrieval-augmented foundation laid out in Section 3.2.1 is particularly relevant to case-based reasoning. Wiratunga et al. (2024) developed CBR-RAG, which enhances legal-domain LLM queries by organising the non-parametric memory so that previous judicial decisions can be matched effectively to incoming queries (Wiratunga et al. 2024, pp. 1–2). This partly addresses the black-box and source-attribution problems of plain LLMs, where it is unclear which legal or factual material informs a given answer (Wiratunga et al. 2024, p. 3). Each chunk is represented by a neural embedding; the authors compared BERT, AnglEBERT, and LegalBERT as encoders (Wiratunga et al. 2024, pp. 6–8). AnglEBERT is itself built on top of BERT embeddings through a contrastive learning step over positive and negative text pairs, in the manner of a Siamese network (Wiratunga et al. 2024, p. 9). The retrieval itself uses two complementary embedding strategies: *intra-embeddings* for attribute-to-attribute comparison, such as matching a query question to stored questions, and *inter-embeddings* for cross-attribute retrieval, in which a query may instead be compared to the supporting text or the entities of stored cases. Intra-embedding retrieval applies a function $f(\cdot)$ to the embeddings; the best matching case is identified by

$$\beta_k = \underset{c_i \in C,\, k}{\text{top-k}}\ \text{Sim}(f(\mathcal{Q}), f(Q_i)) \tag{3.1}$$

Inter-embedding retrieval applies a separate function $g(\cdot)$ that relaxes the strict attribute-to-attribute requirement, giving

$$\beta_k = \underset{c_i \in C,\, k}{\text{top-k}}\ \text{Sim}(g(\mathcal{Q}), g(X_i)) \tag{3.2}$$

The two can also be combined into a hybrid weighted retrieval:

$$\beta_k = \underset{c_i \in C,\, k}{\text{top-k}} \; (w_1 \cdot \text{Sim}(f(\mathcal{Q}), f(Q_i)) + w_2 \cdot \text{Sim}(g(\mathcal{Q}), g(S_i)) + w_3 \cdot \text{Sim}(g(\mathcal{Q}), g(E_i))) \tag{3.3}$$

(Wiratunga et al. 2024, pp. 6–7). The three encoders were evaluated empirically: the hybrid configuration with AnglEBERT and $k = 3$, with the full case included in the prompt, yielded the highest cosine score (Wiratunga et al. 2024, pp. 13–14).

LegalCheck. More recently, Van der Meer and Rossi (2026) developed a retrieval-augmented generation system for drafting preliminary legal advice letters in citizens' objection cases for the Municipality of Amsterdam. They couple a large language model with a curated knowledge base of prior cases, legislation, policy, and guidelines (Van der Meer and Rossi 2026, §§1, 3.2). On top of the standard RAG pipeline, the system introduces a Context-Augmented Generation (CAG) layer implemented through multiple stages of LLM prompts, with a human-in-the-loop review of the drafted letters (Van der Meer and Rossi 2026, §§1, 2.2). At a high level, the system ingests the substantive case documentation, the citizen's objection letter and the enforcement officer's report of the specific violation, and produces a textual query summarising the key facts and legal issue. The query is encoded with OpenAI's `text-embedding-ada-002` and matched against the knowledge base. Documents in the knowledge base are pre-processed into semantically meaningful chunks demarcated by section, echoing the segmentation step of LEE (Li et al. 2025), and stored with their embeddings (Van der Meer and Rossi 2026, §3.1). Retrieval returns between 50 and 200 similar objection sections by cosine similarity, after which the corresponding explanations, documenting how jurists handled comparable cases in the past, are pulled and concatenated into a structured prompt (Van der Meer and Rossi 2026, §4.2). The system achieves recall of 93%, precision of 91%, and an F_1 score of 92% (Van der Meer and Rossi 2026, §4.4).

3.2.3 Limitations

Hallucination and miscalibration. LLMs are prone to producing falsehoods, known as *hallucination*, since there is no mechanism within the model to ensure that the generated text is actually correct (Jurafsky and Martin 2026c, p. 22). They are also poorly calibrated: the model's stated confidence in an assertion does not reliably track the probability that the assertion is true (Jurafsky and Martin 2026b, p. 2). This problem is routinely flagged in applied work, both as something to mitigate and as something to remain aware of (Li et al. 2025, p. 1372) (Törnberg 2024, p. 2). Retrieval-augmented generation partially mitigates the issue by supplying the model with correct domain-specific information at query time, but it does not address the underlying shortcoming of LLMs (Jurafsky and Martin 2026c, pp. 22–23) (Jurafsky and Martin 2026b, p. 2) (Gao et al. 2024, p. 1). More broadly, the non-deterministic outputs and opaque internal reasoning of LLMs sit uneasily with the ideals of transparency and traceability that underpin legal reasoning (Hude 2025, pp. 3–4).

Sycophancy and bias. Large language models are also known to be *sycophantic*: they tend to agree with the user beyond what is warranted and to produce flattering answers. When a user states something that is plainly incorrect, the model may agree with it rather than correct the user, which can reinforce false beliefs and tunnel vision (Jurafsky and Martin 2026c, p. 23). LLMs can in addition reinforce, and sometimes amplify, biases present in their training data, a particular concern in the legal domain (Jurafsky and Martin 2026c, p. 25).

The mitigation proposed in this thesis is to constrain the large language model with an external, symbolic structure of rules, and to apply case-based reasoning over a curated set of prior Dutch judicial decisions to handle open-textured elements. Embedding an LLM in a symbolic framework in this way is known as a *neural subroutine*, and is explored further in the next section. As Hude (2025, p. 4) notes, neuro-symbolic AI may help to overcome the limitations discussed above.

3.3 Neuro-symbolic AI

3.3.1 Theoretical foundation

The debate between the two approaches set out above, neurons versus symbols, has been running for decades (Garcez and Lamb 2023, p. 12388). As early as 2003, Leslie Valiant identified combining reasoning and learning into a single intelligent cognitive behaviour as the main challenge in this debate: 'the ability to learn from experience and the ability to reason from what has been learned' (as cited in Garcez and Lamb 2023, p. 12389). It is sometimes argued that purely neural approaches reach a plateau, because human-like AI ultimately requires symbolic reasoning, just as humans rely on it for complex reasoning tasks (Kautz 2022, p. 118). At the centre of neuro-symbolic AI, accordingly, is the construction of symbolic and neural modules that communicate with one another in a more integrated way (Garcez and Lamb 2023, p. 12391), combining the strengths of symbolic AI, namely rule-based reasoning and logic, with the capacity of neural networks to learn from large datasets (Bhuyan et al. 2025, pp. 8–9). Integrating the two is not straightforward, however; a recurring question is how symbolic and neural systems are to communicate seamlessly (Bhuyan et al. 2025, p. 10). The possible answers to this question have been organised into a taxonomy of integration types by Henry Kautz (Bhuyan et al. 2025, pp. 165–166).

Kautz's taxonomy. Kautz (2022, p. 118) proposed a taxonomy of the different methodologies for neuro-symbolic integration, offering a framework for combining the two approaches effectively (Bhuyan et al. 2025, p. 166). The six designs he identifies are stated briefly below, after which two are examined in more detail.

1. **Symbolic Neuro Symbolic**: symbolic inputs, such as sequences of words in natural-language processing, are converted to vectors, passed to a neural network, and the output is converted back into a symbolic category or sequence.
2. **Symbolic[Neuro]**: a symbolic problem solver acts as the primary reasoning engine and calls a neural pattern-recognition subroutine.
3. **Neuro|Symbolic**: a neural network converts non-symbolic input into a symbolic data structure that is then processed by a symbolic reasoning system. This resembles Symbolic[Neuro], except that the neural part is a coroutine rather than a subroutine.
4. **Neuro: Symbolic → Neuro**: a neural system is given a special training regime based on symbolic rules, absorbing those rules so thoroughly that it can guess the correct answers.
5. **Neuro$_{\text{Symbolic}}$**: symbolic rules are transformed into templates for structures inside the neural network, as in tensor product representations and logic tensor networks.
6. **Neuro[Symbolic]**: a symbolic reasoning engine is embedded within a neural engine, reflecting the idea that human cognition combines a fast, intuitive mode (System 1) with a slow, deliberative one (System 2); this connection is developed below.

(Kautz 2022, pp. 118–119) (Bhuyan et al. 2025, p. 166). Of these six designs, the two that are computationally feasible for the present work are Symbolic[Neuro] (Type 2) and Neuro|Symbolic (Type 3), which couple a neural component to a symbolic engine as a subroutine and as a coroutine respectively; both are examined below and employed in this thesis.

Type 2: Symbolic[Neuro]. Here the primary engine is symbolic and is loosely coupled with neural subroutines that handle specific tasks which cannot be addressed symbolically. The example most often cited is AlphaGo, in which a Monte Carlo tree search calls a neural network for its heuristic evaluation function (Bhuyan et al. 2025, p. 171) (Kautz 2022, p. 118) (Garcez and Lamb 2023, p. 12396). The symbolic problem solver is responsible for high-level reasoning and may be implemented with search algorithms such as Monte Carlo, rule-based systems, or other logic-based systems. Tasks that require learning or pattern recognition are offloaded to a neural

subroutine, and an interface layer ensures that data flows between the symbolic and neural components, so that they operate in harmony as a single neuro-symbolic system (Bhuyan et al. 2025, p. 172). The advantages are modularity per subroutine, data efficiency (high performance is achievable with less data), and the interpretability inherited from the symbolic core. Its limits are implementation complexity, the lack of modularity of the primary symbolic solver, which is largely static, and the difficulty of transforming data and communicating between the two paradigms (Bhuyan et al. 2025, pp. 173–174).

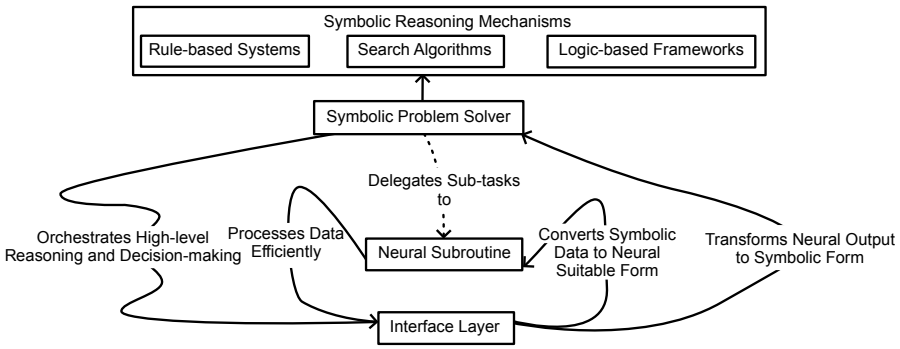


Figure 3.1: The Symbolic[Neuro] (Type 2) architecture: a symbolic problem solver calling a neural subroutine (Bhuyan et al. 2025, p. 172).

Type 3: Neuro|Symbolic. This type is closely related to the previous one, but the division of labour is sharper: a neural network handles one task, such as object or pattern detection, and interacts through its inputs and outputs with a symbolic system specialising in a complementary task, so that the two work in tandem (Bhuyan et al. 2025, p. 174) (Garcez and Lamb 2023, p. 12396) (Kautz 2022, pp. 118–119). Neural networks can also refine symbolic representations: in program synthesis, for instance, deep-learning models generate the symbolic programs and rule systems that satisfy a high-level task specification (Bhuyan et al. 2025, p. 174). The main advantage is specialisation, each system doing what it does best, together with adaptability across a wide range of tasks; the main limit is the complexity of the interaction between the neural and symbolic components (Bhuyan et al. 2025, p. 176).

Legal reasoning as a dual-process structure. Kautz (2022, p. 119) develops the idea that neuro-symbolic AI is closely aligned with the System 1 and System 2 thinking described by Kahneman in *Thinking, Fast and Slow*. System 1 operates quickly and automatically with little effort, whereas System 2 allocates attention to demanding activities, and System 1 calls on System 2 for difficult tasks (Kahneman 2011, pp. 20–21, 24). Guthrie, Rachlinski, and Wistrich (2007, pp. 6–9) extend this dual-process model to judicial decision-making: judges form an intuitive initial judgement (System 1) that deliberation may or may not override (System 2), combining intuitive reaction with deliberative thought. Taken to its conclusion, this view of legal reasoning calls for a Type 6 Neuro[Symbolic] architecture; no full-fledged system of that kind yet exists, however (Kautz 2022, pp. 119–120) (Bhuyan et al. 2025, pp. 183–184) (Garcez and Lamb 2023, p. 12397). This thesis therefore adopts the inverse, more tractable coupling: a neural component embedded in a symbolic framework. Hude (2025, pp. 3–4) grounds this division of labour in H.L.A. Hart's theory of adjudication, assigning symbolic reasoning to the 'easy cases', where the legal rules are clear and determinate, and a neural model to the 'hard cases', where the rules 'run out' and judicial discretion is required. It fits the hybrid design that Garcez and Lamb (2023, p. 12398) call Option 2, 'a more hybrid approach ... whereby the network interacts with a symbolic system for reasoning'.

Explainable AI (XAI). Many scholars stress that explainable AI is essential in sensitive domains. Bhuyan et al. (2025) note that the opacity of out-of-the-box LLMs and neural systems is a significant obstacle to legal applications, where clear explanations are required: an opaque reasoning process is difficult to apply correctly to law, since how a decision was reached remains

largely untraceable (Hude 2025, pp. 3–4). Neuro-symbolic systems can supply this explainability (Bhuyan et al. 2025, pp. 185–186). A knowledge base of cases used for case-based reasoning offers traceability, in contrast to a neural system whose knowledge origin remains unknown (Garcez and Lamb 2023, p. 12397).

3.3.2 Applications in legal domains

The applications surveyed here remain comparatively few: neuro-symbolic AI in law is an emergent and fast-moving field, and only a handful of concrete systems exist. Three illustrative examples are discussed below, spanning three decades.

The Split-Up system. The first initiative to apply neuro-symbolic AI in the legal domain dates back to 1995: the Split-Up system of Zeleznikow and Stranieri (1995). The system addresses the Australian Family Law Act 1975, which governs divorce, children, and property. The authors already observed that neural networks cannot explain their conclusions, and that this 'black-box' character makes it difficult to apply what is learned from a dataset (Zeleznikow and Stranieri 1995, pp. 185–186). The reasoning steps are therefore implemented symbolically, with learning from data done neurally: 'We suggest that useful systems can be built in legal domains which seem, at first sight, too vague for a symbolic approach and too complex for connectionism, if the task is decomposed carefully.' The system uses independent modules, each resolving a sub-task on its own and passing its result on to the next until the whole problem is solved (Zeleznikow and Stranieri 1995, p. 187). Their approach assumed that the explanation a legal system requires can be produced independently of the reasoning itself: a neural network first reaches a conclusion, and a separate explication system then generates an explanation to support it. This choice must be seen in the context of the early state of neural networks at the time. Toulmin argument structures were used to infer the conclusions and to generate plausible explanations for them afterwards (Zeleznikow and Stranieri 1995, pp. 188–189). The work is significant in showing that decomposing a problem into sub-tasks, and using a neural network for sub-tasks within a symbolic framework, can produce the explainable AI that the legal domain demands (Zeleznikow and Stranieri 1995, p. 193).

Guiding LLMs with logic. Kant et al. (2025) demonstrate a modern approach to integrating LLMs with logic programming, in which the terms of insurance contracts are transformed into logic programs whose evaluation outperforms vanilla LLMs on legal queries about the contract (Kant et al. 2025, p. 2). The sometimes unreliable output of vanilla LLM approaches raises doubts about their adequacy for high-stakes legal reasoning (Kant et al. 2025, p. 3), which motivates a guided pipeline that generates Prolog encodings of a policy. The LLM is prompted with the text of a policy and the facts of a claim, and produces a computable contract in Prolog. Of the four models tested, three performed well, GPT-4o, OpenAI o1, and OpenAI o3-mini, with o1 best across every approach, vanilla, unguided, and guided (Kant et al. 2025, p. 7). The most instructive finding concerns the role of guidance: the *unguided* formalisation, in which the model was asked to encode the contract in Prolog with no scaffolding, produced ambiguous, inaccurate encodings that performed *worse* than the vanilla LLM, whereas the guided approach reached full accuracy and consistency with GPT-4o and o1. The benefit of the symbolic layer thus depends on the formalisation itself being carefully guided rather than left to the model.

Structured prompting for rule application. Sadowski and Chudziak (2025) likewise find that, while LLMs can be powerful in the legal domain, they struggle with consistent rule application, transparent reasoning, and explainable outcomes. Their approach is a three-step process consisting of term identification, predicate extraction, and rule application through symbolic verification, designed to combine the strengths of symbolic reasoning and neural learning and to recognise when a general rule does not apply because of special conditions (Sadowski and Chudziak 2025, pp. 1–2). The first step, term identification, extracts from unstructured text the entities corresponding to terms in the reasoning task. The second, predicate extraction, determines which predicates apply to those terms: the LLM analyses the relationships between them and so captures the relational aspect of reasoning. The third, rule application, applies formal logic to decide

whether the task predicate holds given the extracted predicates, expressing the logic as formulae verified with an SMT solver (Sadowski and Chudziak 2025, p. 4). They apply the framework to the determination of hearsay, formalised in first-order logic. The study uses complementary predicates (for example both `IsInCourt` and `IsOutOfCourt`), since LLMs exhibit a confirmation bias towards positive instances and neglect their negations (Sadowski and Chudziak 2025, p. 5). Evaluated across six models from three providers, the framework improved most consistently on OpenAI's o-series: both o1 and o3-mini gained substantially, with o1 reaching an F_1 score of 0.929 under the structured prompting that integrates neural and symbolic approaches, compared with 0.74 under few-shot prompting alone (Sadowski and Chudziak 2025, pp. 7–9).

These examples converge on a single design for the present research: a symbolic model of Article 42 Fw, the rule-based skeleton of the actio pauliana, coupled with neural subroutines that resolve its open-textured elements against a curated body of Dutch case law. The remainder of this thesis develops and evaluates such a system.

CHAPTER 4

Method

This chapter develops the neuro-symbolic system for the actio pauliana whose foundations were laid in Chapter 3. It begins with the architecture (Section 4.1) and the symbolic core that forms the primary system (Section 4.2). Because that core deliberately leaves a few open-textured elements undefined, the rest of the chapter describes what fills them: a case base built from Dutch case law (Section 4.3) and the neural subroutines that query it (Section 4.4).

4.1 Architecture

The system couples a symbolic problem solver with neural subroutines, following the Type 2 Symbolic[Neuro] and Type 3 Neuro|Symbolic designs of Section 3.3.1. The symbolic core is a machine-readable model of Articles 42, 43 and 45 Fw that, given input variables describing a contested transaction, decides whether it is voidable (*vernietigbaar*). It drives the neuro-symbolic model and the neural components are subordinate to it.

A neural component couples to this core at two points. On the input side, the model's variables must be retrieved from the unstructured facts of a case: a large language model converts a natural-language description of a transaction into the structured (JSON) inputs the model expects. Because it turns non-symbolic input into a symbolic structure for a symbolic reasoner, this is closely aligned to Kautz's Type 3 Neuro|Symbolic design and could be described as a neural coroutine (Kautz 2022, pp. 118–119).

Inside the model's reasoning, a few elements are open-textured, cannot be reduced to a rule and are developed in case law. These are where, in the terms of Chapter 3, the rules 'run out'. A neural subroutine resolves an open-textured term through case-based reasoning over retrieved prior decisions, returning a value and a justification grounded in those decisions; as the symbolic solver remains the primary system and calls the neural component only here, this is a Type 2 Symbolic[Neuro] design (Kautz 2022, p. 118). That subroutine needs a curated body of Dutch case law, and building this case base, i.e. collecting, surveying and segmenting the judgments, is itself a substantial part of the method (Sections 4.3–4.4).

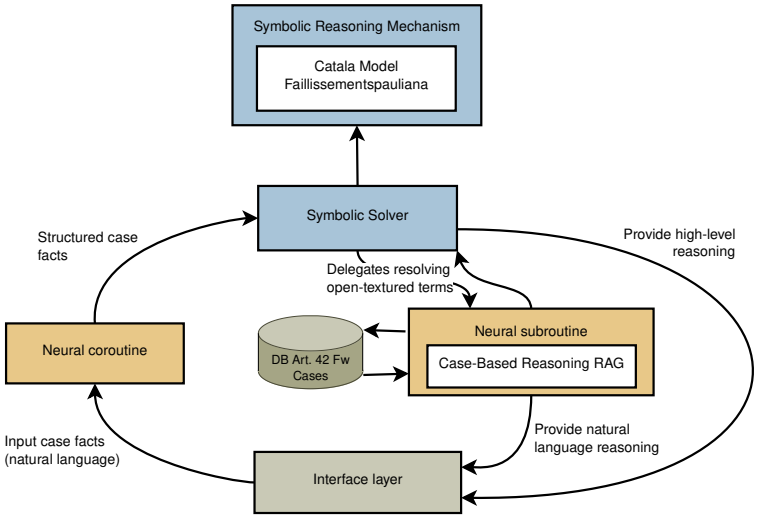


Figure 4.1: High-level overview of the architecture. The symbolic solver with the Catala model of the actio pauliana is the primary system and delegates the resolution of open-textured terms to a neural subroutine that performs case-based reasoning over the case base, while a neural coroutine structures the natural-language facts of a case into the model's inputs.

4.2 A symbolic model of the actio pauliana

The model is written in Catala rather than a general logic-programming language such as Prolog. Prolog has long been the default for symbolic law, as the British Nationality Act experiment of Chapter 3 began to show. Catala was designed to translate legislation into executable code while keeping a one-to-one correspondence between a legal paragraph and its formalisation (Merigoux, Chataing, and Protzenko 2021, p. 3). Its decisive advantage here is methodological: Catala is built around literate and pair programming, a computer scientist and a legal scholar annotating the statute side by side (Huttner and Merigoux 2022, §4.2.1). The model was developed similarly, the source text of Articles 42, 43 and 45 Fw was paired with the formalisation. Catala also compiles to Python (Merigoux, Chataing, and Protzenko 2021, p. 3), which makes calling the neural subroutines from the model straightforward.

Building the model follows the normalisation Allen and Engholm (1978) described for statutes: atomic propositions are identified, replaced by labels, made explicit with propositional logic, and the original text restored to validate the result (Ashley 2017, pp. 41–43), in predicate form where possible (Ashley 2017, pp. 44–45). The evidentiary presumptions are modelled as subscopes the Article 42 scope invokes: Article 45 Fw (the presumption of knowledge for acts for no consideration) and Article 43 Fw (the presumptions for legal acts for consideration).

A decision is taken about where the symbolic model stops. Some elements require more complex reasoning, such as whether an act is voluntary, the prejudice to creditors consisting of the hypothetical comparison of the estate with and without the transaction, and the knowledge requirement defined by an objectified foreseeability test, developed in case law. Rather than force them into rules, the model takes them as boolean *inputs* marking the seams where a neural subroutine couples to the engine. The source annotates each input accordingly.

The model follows the standard decision scheme of the actio pauliana and makes a few deliberate simplifications. None of these affects the open-textured elements that the neural subroutines resolve. The final Catala model is listed in Appendix D.

4.3 Corpus construction

The empirical component of this research relies on a corpus of judicial decisions concerning the actio pauliana under Article 42 Fw. All judgments are sourced from rechtspraak.nl, the official

open-data publication channel of the Dutch judiciary (IVO Rechtspraak 2019). Rather than querying the Rechtspraak search interface, a complete metadata dump of published decisions is obtained from WooZM, the Woo ZoekMachine maintained by the ICAI OpenGov Lab. Each record in this dump contains a unique European Case Law Identifier (ECLI), together with the Dublin Core metadata fields exposed by the Rechtspraak schema, including title, description, and legal-domain classification.

The corpus is constructed in two stages. In the first stage, the WooZM metadata dump is filtered on the legal-domain classification to retain only decisions in the civil-law domain. The associated ECLIs are then used to retrieve the corresponding judgment texts through the `content` endpoint of the Rechtspraak Open Data API, which returns each decision as an XML document conforming to the schema described in Chapter 2. This yields a working set of over 270,000 civil-law XML files, which constitutes the base corpus for this research.

In the second stage, a regular-expression matcher is applied to the full text of every XML file in the base corpus. For each of Articles 42, 43 and 45 Fw a single case-insensitive pattern is assembled from three parts: a prefix matching the various forms of *article*, a bounded gap, the article number under word boundaries, a second gap, and a suffix matching the statute. With `ARTICLES = [42, 43, 45]`, the three patterns are built as:

Listing 4.1: The article matcher. The `0,30` gap lets the number sit a short distance from *artikel* and *Fw*, and `artt?\.?` also admits the plural abbreviation *artt.* (e.g. *artt. 42–45 Fw*).

```
PREFIX = r"(?:artt?\.?|artikel)\b"
GAP    = r"[\s\S]{0,30}?"
SUFFIX = r"(?:Fw|Faillissementswet)\b"
```

This matches surface forms such as *artikel 42 Fw*, *art. 43 Faillissementswet* and *artt. 42 en 45 Fw*. Deliberately, no keyword condition on *pauliana* or *paulianeus* is added: those terms also denote the civil-law actio pauliana of Article 3:45 BW, which falls outside the scope of this research, so filtering on them would admit many false positives. Articles 43 and 45 Fw are included alongside Article 42 Fw because they introduce the evidentiary presumptions discussed in Chapter 2, and decisions invoking these presumptions are substantively part of the actio pauliana case law even where Article 42 Fw is not itself cited. Operating directly on the body of each judgment rather than on the often incomplete `dc_description` field avoids the recall limitations of metadata-only filtering. Finally, only decisions whose ECLI court code begins with `RB` (District Court), `GH` (Court of Appeal) or `HR` (Supreme Court) are retained, while the conclusions of the procurator general at the Supreme Court (`PHR`) are excluded, as these are advisory opinions rather than judgments. This yields a working corpus of 846 XML documents for the corpus survey and segmentation pipeline described in the following sections.

4.3.1 Corpus survey

The structural irregularity of Rechtspraak XML documented in Section 2.3 has motivated several attempts at automated section assignment, but the efforts so far are less suited to the section segmentation required here. Trompper and Winkels (2016) labels text elements with Conditional Random Fields and parses the tagged sequence into a section tree with a Probabilistic Context-Free Grammar and a CYK parser, reporting an F_1 around 0.91. The approach works even on poorly marked-up text, but the website, database and training file are now offline, and retraining an unmaintained 2016 Java codebase against XML that has since evolved offers no guarantee of success. The more recent open implementation by Crombach, Chakravarthy, and Wicker (2025) runs after some dependency fixes, but its segmentation is content-agnostic: where role attributes are absent it falls back to sentence-level chunking over trigram windows and stopwords rather than identifying section boundaries, and its tag-based path keys only on title and number tags, which the flat older judgments frequently lack. Content-agnostic chunking, splitting mid-section, is precisely what this research seeks to avoid.

Rather than train a new classifier, a Large Language Model is used to *survey* the corpus rather than to annotate it directly. The model produces a JSON report cataloguing the surface forms under which each target section appears; this report is then used to develop a deterministic

segmenter of Section 4.3.2. The idea is that the variability Tjong Tjin Tai (2021, p. 169) describes, headings such as 'De beslissing' and 'rechtdoende' all marking the dictum, is finite but unknown in advance, and a rule set is preferable once it is known what the corpus actually contains.

Törnberg (2024) lists six factors for choosing a large language model: reproducibility, ethics and legality, transparency, language fit, scalability, and complexity. Four carry weight here. Reproducibility and transparency favour an open-weight model pinned to a fixed release; scalability matters because the prompt is iterated several times over nearly 850 files, so a full pass must take minutes rather than days; and the task is of modest complexity, the model only locating known sections rather than reasoning openly. On this basis, Mistral Small 3.2 (24B parameters, instruct-tuned, release `mistral-small-3.2-24b-instruct-2506`) is used through the Scaleway Generative APIs. Using an API rather than local inference reflects the scalability requirement, since a 24B model can be difficult to run on consumer hardware. Inference uses `temperature=0.1`, `top_p=0.2`, and `seed=42`; temperature is not set to zero because Scaleway documents a risk of infinite generation loops when zero temperature is combined with structured output. The prompt itself was refined with assistance from a separate model (Claude Sonnet 4.6 via `aichat.uva.nl`).

A single system prompt lists the five target sections, enumerates the common title variants, and gives few-shot examples. It distinguishes sections identifiable from the XML `role` attribute (*procesverloop*, *overwegingen*, *beslissing*) from those that must be inferred from the title alone (*feiten*, *vorderingen*), which have no role value in the schema; *feiten* is recorded only when separately headed, an integrated factual recital inside the *overwegingen* being left null. Output is constrained by a Pydantic schema: for each document the model returns the five string or null fields, one per section, each holding the opening tag and title that marks the section's start. This removes the need to parse free-form output and makes aggregation trivial.

A training set is available without manual labelling, as recent judgments already carry `role` attributes on the three populated sections, which serve as gold-standard examples. Validation, however, departs from Törnberg (2024). Rather than checking output labels against a held-out sample, the survey output helps to build a fixed rule set, so validation takes the form of outlier inspection on the aggregated JSONL. Rare or atypical section openings are reviewed against the source; those reflecting genuine drafting variation are absorbed into the rule set, those reflecting misclassification discarded. The rule set's coverage is thus justified by the survey, but its precision does not depend on the model being correct on every document.

4.3.2 Deterministic segmentation

The corpus survey helps to establish the finite repertoire of surface forms under which each section appears. A segmentation system then assigns every part of a judgment to one of the canonical sections deterministically. Splitting a judgment along these meaningful boundaries, rather than at fixed token offsets, is what makes the downstream retrieval system effective, since it keeps logical units intact (Section 3.2.1) (Gao et al. 2024, pp. 7–8). Rule-based segmentation is a well-established and effective choice for this in the legal domain (Li et al. 2025) (Jackson et al. 2003).

The segmenter is built as a short cascade of finite-state transducers, taking its design from FASTUS, the cascaded finite-state system for information extraction (Hobbs et al. 1993, p. 133) (Jurafsky and Martin 2026a, p. 19). However, it also differs, as FASTUS runs over a tape of word tokens, whereas the segmenter runs over a tape of XML block elements (the `para`, `title` and `bridgehead` nodes inside the `uitspraak` element) in document order. Because the input is already marked-up XML, the lowest FASTUS stages, which recognise tokens and basic phrases, are effectively performed by the markup, and the segmenter starts higher up the cascade.

At the heart of the labelling is a lexicon that is finite-state. An ordered list of (regular expression, label) pairs is applied to each heading and the first pattern to match wins; the ordering resolves overlaps by fixed priority. The phrase *De beoordeling van het geschil*, for instance, is tested against the *overwegingen* pattern before the *geschil* pattern, so that it is labelled by its leading word rather than by the noun *geschil*. A second mechanism gates which lines may become headings: a line is promoted only if it is short and built entirely from a closed vocabulary of heading words. A running sentence such as *Het verweer slaagt niet* contains the section keyword

verweer but also the off-vocabulary word *slaagt*, and is therefore refused; it is body text, not a heading. This gate embodies the FASTUS principle of robustness through doing the right amount of recognition and declining to over-interpret (Hobbs et al. 1993, p. 137). The algorithm itself is court-independent; the label patterns and the heading vocabulary form the domain-dependent component, selected from the court code in the ECLI.

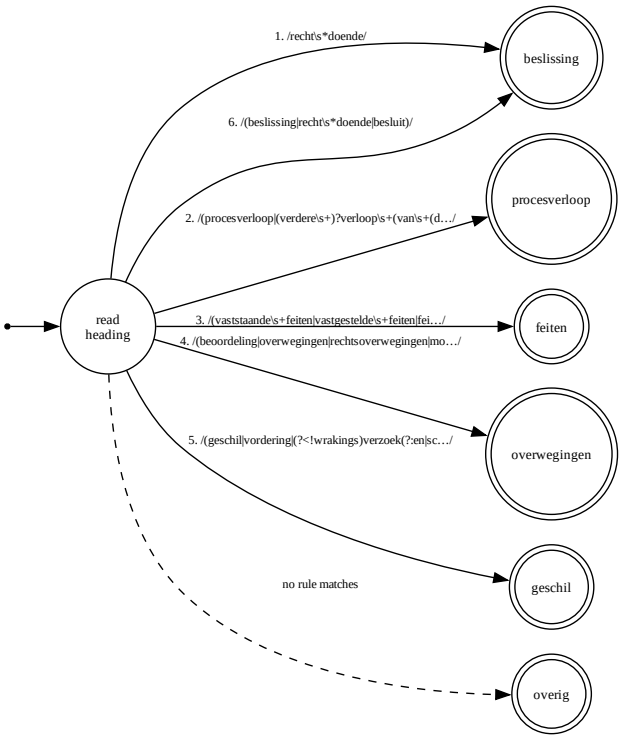


Figure 4.2: Example of the district court lexicon rules and labels: the finite-state label transducer mapping a heading to the section. These are priority-ordered.

The pipeline can be viewed as a three-level feeding cascade, each level consuming the previous level's output. *Recognition* emits labelled section markers and is itself a priority-ordered disjunction of four recognisers, corresponding to four layout eras of Rechtspraak documents: `structured` (top-level `section` markup), `bridgehead` (emphasis headings), `numbered-para` (numbered lines such as *3. Het geschil*), and `keyword` (bare heading lines in the oldest, flattest documents). A recogniser wins only if it yields at least one canonical section; otherwise the next is tried. *Refinement* then recovers structure the front end missed (a leading *procesverloop*, the closing *beslissing*, or an oblique facts heading such as *De kern van de zaak*) and drops false structures, such as the party caption preceding the first real section or a quoted line mistaken for a heading. *Closing*, finally, pairs each start marker with the next to form spans and collects the text between them, turning the markers into a partition of the document into labelled sections.

The segmenter returns the labelled spans of a judgment. These spans are the semantic chunks consumed by the retrieval stage of Section 4.4. It is validated in the way standard for finite-state extraction, by recall and precision against a manually reviewed sample: the documents whose segmentation is wrong are marked manually. The reported measures are recall, precision and F_1, in line with the segmentation results reported elsewhere in the legal-NLP literature (Van der Meer and Rossi 2026, §4.4).

4.4 Neuro-symbolic model

As set out in Section 4.1, the symbolic model couples to neural routines: a neural coroutine that structures the case facts in natural language, and neural subroutines that resolve open-textured elements.

The first transforms the unstructured facts of a contested transaction into the structured inputs of the Catala model. A large language model is prompted with the facts of a case and returns the boolean and date fields of the main scope, such as the date of the act, whether it is a legal act performed by the debtor and so on, as a structured JSON object, in the same constrained-output manner used for the corpus survey.

Neural subroutines resolve the open-textured terms, prejudice to creditors, the distinction between voluntary and due performances, and the knowledge requirements (knowledge by the debtor and by the counterparty) by case-based reasoning over the case base. The LLM prompts are listed in Appendix B. For a contested transaction the most relevant prior decisions are retrieved and supplied to a language model, which returns a boolean value together with an explanation grounded in those decisions, satisfying the explainability requirement that Section 3.3.1 treats as non-negotiable for legal applications. Retrieval follows the embedding-based approach of Section 3.2.1: a segmented chunk, facts of a case from the Rechtspraak corpus, is encoded as a vector and matched to the query by cosine similarity. Embeddings are computed with `qwen3-embedding-8b`, a multilingual model with a 32k-token context window, served through the Scaleway Generative APIs. The long context is material here, as it allows an entire reasoning section of a judgment to be embedded as a single chunk, well beyond what an encoder in the BERT family admits (Section 3.2.1). The chunks and their embeddings are stored in a PostgreSQL database, against which retrieval queries are run.

Retrieval scores a judgment against the query through a hybrid, weighted combination over two attribute types, the facts of the case (F) and the court's reasoning (*overwegingen*, O), taking for each attribute its best-matching chunk in the manner of the *maxP* passage aggregation of Dai and Callan (2019) and weighting the attributes that are present by w_t:

$$\text{score}(q, d) = \frac{\sum_{t \in \{F,O\}} 1[S_t(d) \neq \emptyset]\ w_t \max_{\mathbf{s} \in S_t(d)} \text{sim}(q, \mathbf{s})}{\sum_{t \in \{F,O\}} 1[S_t(d) \neq \emptyset]\ w_t} \tag{4.1}$$

The choice of the embedding model is a delimitation rather than a claim of what is best for case-based reasoning. A domain-adapted or fine-tuned encoder might improve retrieval over the specialist vocabulary of insolvency law (Section 3.2.1). Within the scope of this thesis a readily available and sufficiently capable embedding model that keeps the retrieval pipeline tractable is preferred, so that effort can be concentrated on the symbolic model and on the case base the subroutine depends upon.

Following the pre- and post-retrieval stages of Gao et al. (2024, pp. 3–4), two optimisations concerning the substantive legal criteria and the procedural chain of a case are introduced. The pre-retrieval stage narrows the candidate set to decisions that are actually relevant to the subroutine at hand: each judgment in the case database carries a boolean label describing whether it in fact rules on the element in question, prejudice to creditors, the distinction between voluntary and due performances, or one of the knowledge requirements, and only matching decisions are searched. To select the model that produces these labels, a gold standard of 100 judgments was manually annotated for the three booleans, and two candidate models, `gpt-5.4` and `mistral-medium-3.5-128b:fp8`, were evaluated against it. The coding procedure used is outlined in Appendix A. The labels of `mistral-medium-3.5-128b:fp8` were adopted for the full corpus: its recall was slightly higher (Table 5.2), which matters most here since missing a relevant ruling is costlier than retrieving an irrelevant one, and, perhaps more importantly, the model has open weights and can be run across different inference providers, a real advantage for reproducibility, also in an academic setting.

The post-retrieval stage expands the result set along the procedural chain: when a retrieved decision belongs to a case that was also litigated at another instance, the decisions from those

other instances are added as well. A retrieved district court (`RB`) judgment is supplemented with any court of appeal (`GH`) or Supreme Court (`HR`) decision that follows it, and a higher-instance decision is supplemented with the lower-instance judgments that precede it. In both directions the original is retained. This is worthwhile because courts at a higher instance frequently rely on facts established below without restating them, so a lower-instance decision may supply context the higher one omits, while a lower-instance decision may have been overruled on appeal, so the higher-instance decision is needed to establish the final reasoning and decision.

The inference parameters used for the large language models across the pre-retrieval annotation and the coroutine and subroutines of the final neuro-symbolic model are listed in Table 4.1.

Table 4.1: Inference parameters for the large language models used in the pre-retrieval annotation and in the coroutine and subroutines of the final neuro-symbolic model. Parameters not listed were left at the provider's default.

Model	Parameters
`mistral-medium-3.5-128b:fp8` (Scaleway), both parts	Temperature: 0.1 Top-p: 0.2 Top-k not set Seed: 42
`gpt-5.4` (OpenAI), pre-retrieval annotation	Seed: 42 Reasoning effort: medium
`gpt-5.5` (OpenAI), neuro-symbolic model	Seed: 42 Reasoning effort: medium
`gemini-3.5-flash` (Gemini, OpenAI-compatible endpoint), neuro-symbolic model	Temperature: 0.1 Top-p: 0.2 Reasoning effort: medium

CHAPTER 5

Evaluation

This chapter reports the empirical evaluation of the experimental setup developed in Chapter 4. It follows the order of the pipeline itself: first the deterministic segmentation that structures the corpus, then the large-language-model annotation that labels each judgment for the pre-retrieval stage, and finally the complete neuro-symbolic model that decides whether a contested transaction is avoidable. Each component is measured against a manually annotated gold standard, and precision, recall, and F_1 are reported throughout. The final model is evaluated on a deliberately small set of hand-annotated cases, and its results are therefore read with corresponding caution.

Firstly, the deterministic segmentation is evaluated against a manually annotated set of $n = 100$ judgments selected through simple random sampling. As Table 5.1 shows, the segmenter attains high precision, recall, and F_1 across all five sections, with a micro average of 0.98 precision and 0.97 recall; the lowest per-section score is on the *overwegingen* section ($F_1 = 0.94$). The segmentation system was built specifically for judgments concerning the Dutch transaction avoidance under Art. 42 Fw, so these results speak only to the *actio pauliana* of Art. 42 Fw and do not necessarily extend to judgments in other fields of civil law.

Table 5.1: Section segmentation performance on the Rechtspraak corpus, reporting precision, recall, and F_1 per section against a manually reviewed sample.

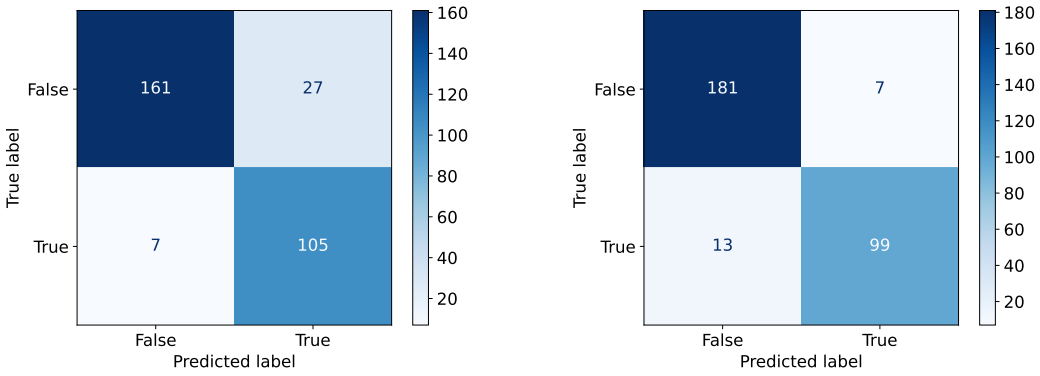
Section	Precision	Recall	F_1	Support
Course of proceedings	1.00	0.99	0.99	100
Established facts	1.00	0.97	0.99	76
Dispute	0.98	0.94	0.96	63
Grounds of the decision	0.95	0.92	0.94	114
Decision	0.99	0.99	0.99	101
Other	0.98	1.00	0.99	128
Micro average	0.98	0.97	0.98	582

The accurate segmentation enables the grounds of the decision, the core reasoning by the court, to be annotated for whether the court rules on one of the three legal criteria. The labels generated by the large language model are evaluated against the same gold standard of $n = 100$, which was also manually annotated for the three labels relevant to the pre-retrieval stage of the retrieval-augmented generation, and are presented in Table 5.2. Precision, recall, and F_1 should all be taken into account, but an important focus here is recall, since for jurists the cost of missing information in a legal context is significantly higher than the cost of reviewing extra, irrelevant results. The two models seem to trade precision off against recall: `mistral-medium-3.5-128b:fp8` reaches the higher recall (micro-average 0.94) at a lower precision (0.80), whereas `gpt-5.4` is the more precise (micro-average 0.93) at a slightly lower recall (0.88), with micro-averaged F_1 scores of 0.86 and 0.91 respectively. Given the emphasis on recall, `mistral-medium-3.5-128b:fp8` misses the fewest relevant rulings, while `gpt-5.4` returns fewer false positives.

Table 5.2: Pre-retrieval annotation performance per substantive element, comparing `mistral-medium-3.5-128b:fp8` and `gpt-5.4` against 100 manually annotated judgments serving as the gold standard. The label records whether the court rules on the element on the grounds of the decision.

Model	Element	Precision	Recall	F_1	Support
`mistral-medium-3.5-128b:fp8`	Voluntary performance	0.69	0.97	0.81	39
	Prejudice to creditors	0.89	0.91	0.90	34
	Knowledge	0.86	0.92	0.89	39
	Micro average	0.80	0.94	0.86	112
`gpt-5.4`	Voluntary performance	0.85	0.90	0.88	39
	Prejudice to creditors	1.00	0.82	0.90	34
	Knowledge	0.97	0.92	0.95	39
	Micro average	0.93	0.88	0.91	112

Further details of how false positives and false negatives relate to true positives and true negatives are provided in the confusion matrices of Figure 5.1. These show that `mistral-medium-3.5-128b:fp8` has a slight positivity bias, predicting the positive label somewhat too readily (relatively more false positives), whereas `gpt-5.4` is more conservative, with relatively more false negatives, similar to what is shown in Table 5.2.



(a) `mistral-medium-3.5-128b:fp8` (b) `gpt-5.4`

Figure 5.1: Confusion matrices for the pre-retrieval annotation of whether the court rules on each element on the grounds of the decision, for the two annotation models.

The final neuro-symbolic model is evaluated against eight hand-annotated cases from outside the `rechtspraak.nl` dataset which are listed in Appendix C. Their facts and the dispute at hand served as the natural-language input for the neuro-symbolic model. Since Catala computes a single output from a given set of inputs, and those inputs consist of the (legal) criteria expressed as booleans, the booleans are compared against the gold standard and the micro-averaged precision, recall, and F_1 are reported for different values of k, relating to the case-based reasoning (retrieval-augmented generation). Because the sample comprises only eight cases, and these cases are themselves legally difficult to judge, the results require caution in interpretation and are reported as indicative rather than conclusive. As Table 5.3 shows, the micro-averaged scores for all three models fall in a comparable range (precision between 0.62 and 0.70, recall between 0.74 and 0.87, and F_1 between 0.67 and 0.75), and for each model the micro-averaged F_1 at $k = 5$ and $k = 10$ is at or above its value at $k = 0$. No ranking between models is drawn from these figures.

Table 5.3: Neuro-symbolic model performance, reporting micro-averaged precision, recall, and F_1 across all boolean criteria and the final verdict over eight judgments against the gold standard, at retrieval depths $k = 0$, $k = 5$, and $k = 10$.

	$k = 0$			$k = 5$			$k = 10$		
Model	P	R	F_1	P	R	F_1	P	R	F_1
`gemini-3.5-flash`	0.62	0.82	0.70	0.63	0.87	0.73	0.65	0.82	0.73
`gpt-5.5`	0.65	0.82	0.73	0.68	0.82	0.74	0.67	0.82	0.74
`mistral-medium-3.5-128b:fp8`	0.62	0.74	0.67	0.70	0.79	0.75	0.69	0.79	0.74

The results can also be split out per neural subroutine, as reported in Table 5.4. The voluntary-performance and prejudice-to-creditors subroutines reach high recall and moderate-to-high F_1 across all three models, whereas the knowledge subroutine attains markedly lower precision, recall, and F_1 throughout. For the knowledge subroutine the micro-averaged scores at $k = 5$ are generally higher than at $k = 0$, so the retrieved prior decisions appear to raise performance on this element, although the same small-sample caution applies.

Table 5.4: Per-subroutine micro-averaged precision, recall, and F_1 at retrieval depths $k = 0$, $k = 5$, and $k = 10$, for each model's three substantive subroutines.

		$k = 0$			$k = 5$			$k = 10$		
Model	Subroutine	P	R	F_1	P	R	F_1	P	R	F_1
`gemini-3.5-flash`	Voluntary performance	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
	Prejudice to creditors	0.62	1.00	0.77	0.62	1.00	0.77	0.71	1.00	0.83
	Knowledge	0.12	0.33	0.18	0.21	0.67	0.32	0.14	0.33	0.20
`gpt-5.5`	Voluntary performance	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
	Prejudice to creditors	0.71	1.00	0.83	0.83	1.00	0.91	0.71	1.00	0.83
	Knowledge	0.14	0.33	0.20	0.15	0.33	0.21	0.15	0.33	0.21
`mistral-medium-3.5-128b:fp8`	Voluntary performance	0.75	1.00	0.86	0.86	1.00	0.92	0.86	1.00	0.92
	Prejudice to creditors	0.62	1.00	0.77	0.62	1.00	0.77	0.62	1.00	0.77
	Knowledge	0.09	0.17	0.12	0.27	0.50	0.35	0.25	0.50	0.33

Finally, the accuracy of the final verdict, whether the transaction is avoidable under Art. 42 Fw, is reported in Figure 5.2 as the number of correct verdicts out of the eight cases. The accuracy is modest for this sample. Manual inspection of the neuro-symbolic model's output reveals that misjudging the knowledge requirement is what usually lands a different final verdict. For all three models the highest number of correct verdicts is reached at $k = 5$ rather than at $k = 10$. It may be noted that the judgments supplied through retrieval-augmented generation are frequently lengthy in terms of words and tokens.

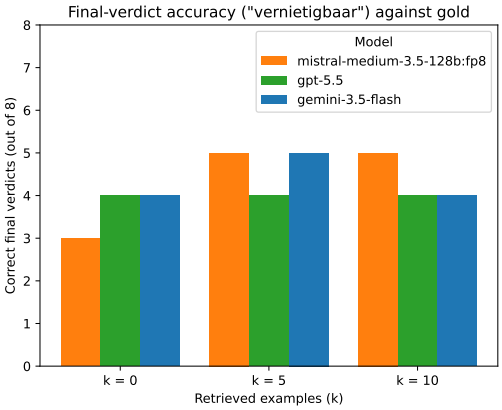


Figure 5.2: Number of correct final verdicts (avoidable under Art. 42 Fw) out of the eight gold-standard cases, per model at retrieval depths $k = 0$, $k = 5$, and $k = 10$.

CHAPTER 6

Conclusion

This thesis set out to investigate how formal legal modelling and data-driven methods can be combined to capture the application of Article 42 Fw, the *actio pauliana* of Dutch insolvency law. It has shown that such a statute can be translated into a symbolic model consisting of different legal criteria, which can be resolved neurally with the help of large language models. The legal structure of Article 42 Fw lends itself to a *law-as-code* representation: after the statutory text is normalised, the different elements are expressed in Catala, a language that keeps a one-to-one correspondence between a legal provision and its formalisation. The evidentiary presumptions of Articles 43 and 45 Fw are modelled as subscopes invoked by the main Article 42 scope, while the open-textured elements that resist reduction to a rule, prejudice to creditors and the knowledge requirement, are left as boolean inputs that mark the seams where a neural component couples to the model. Keeping the formalisation close to the statutory text, and developing it in the literate, pair-programming style that Catala encourages, benefits both the explainability and the accessibility of the resulting model.

In terms of judicial data, processing the publicly available case law from `rechtspraak.nl` required an extensive pipeline of successive cleaning and structuring phases. At its core is a deterministic segmentation system that divides each judgment into sections grounded in Dutch civil procedure law and court drafting practice. The evaluation showed that, within the Article 42 Fw corpus, this segmenter achieves high precision and recall. Reliable segmentation in turn enables the court's reasoning to be annotated against the substantive legal criteria, recording whether the court rules on the voluntary performance, the prejudice to creditors, or the knowledge requirement. Because these annotations can be produced by a large language model, the labelling extends beyond the manually annotated gold standard to the corpus as a whole, yielding the structured dataset on which the retrieval stage depends.

The inputs of the symbolic model can then be resolved neurally through large-language-model (sub)routines. Retrieval-augmented generation was implemented in its more advanced form, with the pre- and post-retrieval stages made possible by the parsing of the judgments: candidate decisions are first narrowed by their substantive labels and then expanded along the procedural chain of a case, storing formal relations between cases along different judicial instances. Case-based reasoning relies on `qwen3-embedding-8b`, with embeddings matched through a hybrid, weighted scheme across the facts and the court's reasoning, set against a novel input of facts and a dispute. Because Catala compiles to Python, the complete neuro-symbolic model can call the neural subroutines from a single Python architecture which consists of both the strict schema used by the language models and the symbolic reasoning of the model. A limitation is that Catala's Python output does not readily expose a trace of its reasoning, so the intermediate application of the rules is difficult to inspect from the Python alone.

The final neuro-symbolic model was evaluated against a small, manually annotated set of cases by comparing the model's resolved inputs and outputs with the manual annotation. Because the validation set is small, it is difficult to draw firm conclusions about the model's capacity for legal reasoning. The first results nonetheless point to particular difficulty in resolving the open-textured knowledge requirement on the part of the debtor and the counterparty. This

difficulty is compounded by the nature of the validation material: the judgments were, as a rule, selected for publication in jurisprudence journals precisely because they raise contested or borderline points of law, and so are not representative of the more routine cases the model could meet in practice. A practical advantage of the neuro-symbolic design is that its inputs remain symbolically adjustable, which could let practitioners refine them to retrieve more accurate legal reasoning, though this possibility is left unexplored.

Taken together, these findings indicate that formal legal modelling and data-driven methods can be combined into a single, explainable model of Article 42 Fw, in which a symbolic core delegates its open-textured elements to neural subroutines grounded in prior case law. How faithfully such a model can reproduce judicial reasoning, particularly on its hardest, open-textured elements, remains a question that a larger validation effort could help to settle.

CHAPTER 7

Discussion

A first set of reflections concerns the evaluation. Obtaining suitable data to validate the final model is difficult for several reasons. The facts and the dispute are often stated only briefly in a judgment, so the natural-language input given to the neuro-symbolic model is correspondingly small and almost certainly omits part of the factual record that was laid out before the court. A more fundamental difficulty is that this research draws no distinction between easy and hard cases, a question that belongs to legal philosophy and is left unexplored here. That distinction matters for validation: the judgments used as a gold standard were published in jurisprudence journals precisely because they are legally interesting, and are therefore skewed towards the harder cases rather than the routine ones an insolvency practitioner would more often encounter. Future work should accordingly prioritise validation data that more closely reflects how such a model could or would be used in practice.

A second limitation is that the study does not assess the natural-language reasoning the neural subroutines produce, nor whether retrieval-augmented generation is genuinely decisive in resolving a case. The legal method is required to judge whether the subroutines make correct use of the retrieved judgments. Since this is a computer-science thesis, that legal analysis is deliberately left out. The results can show that retrieval steers the model in a particular direction, but not that it works as a reliable safeguard against hallucination, which would need the same qualitative assessment. Future work could apply the legal method to the full neuro-symbolic synthesis, both the natural-language considerations and the symbolic decision graph produced by `catala explain` (as shown in Appendix E), and could also move beyond the court-centric setup used here. Rather than feeding in a judgment's facts and dispute to reproduce an objective verdict, a model might be placed in the hands of an insolvency practitioner who steers it to argue a particular side, in a more practice-oriented validation along the lines of LegalCheck (Van der Meer and Rossi 2026). Legal argument has, after all, two sides: a conclusion deemed wrong may simply reflect the opposite side of the case being argued.

At the architectural level, the design pushes several components slightly beyond their intended use. Catala was developed for fiscal legislation such as tax law, not for coupling to neural components, and that coupling still demands a great deal of manual specification, in particular the prompts and the output schemas the language models must follow. The link between the symbolic model and the neural routines is therefore loose. Revising the Catala model means revisiting much of the surrounding neuro-symbolic system, and the literate, law-aligned style that Catala affords on the symbolic side is lost on the neural side. Extending Catala to better accommodate a neuro-symbolic architecture is a promising direction for future work. At a more conceptual level, the architecture inverts the integration that Kautz (2022) envisages as the field's goal: where human cognition mostly runs on fast, intuitive System 1 thinking and occasionally defers to deliberative System 2, this model has a System 2 symbolic core calling System 1 language models. The reasons for this are clear, but it is not how a jurist reasons. Their judgment begins instead from an intuitive sense of justice that deliberation then tests against the rules (Guthrie, Rachlinski, and Wistrich 2007). This is better seen as a deliberate design choice than as a defect, but it marks a meta-level direction that is left unexplored.

What can be claimed with confidence is that the design stays faithful to the theory it builds on, namely to the legal account of Article 42 Fw in Chapter 2 and to the legal-AI literature in Chapter 3. Each of these has been studied separately before. Bringing them together into a single neuro-symbolic model of the *actio pauliana* is the novel approach of this study.

Finally, the ethical and philosophical implications need to be addressed. In terms of data the concerns are limited, since the corpus and the retrieved decisions are, in one form or another, open data. The deeper issue is philosophical: the law-as-code paradigm presumes that applying legal rules can be treated as a broadly mechanical operation, extending even to interpretation and explanation. Whether that premise holds is contested among legal scholars, and this research necessarily adopts the premise. A more concrete ethical risk is that language models can amplify biases, whether present in the supplied material or in their vast training data, and in this setting such amplification can never be fully ruled out (Jurafsky and Martin 2026c). This reinforces a recurring theme: that a tool of this kind is best applied not to judgments in isolation but in support of an insolvency practitioner who steers it and retains the final say on the verdict, much as the professional in LegalCheck fixed the legal outcome in advance (Van der Meer and Rossi 2026). Such a human-in-the-loop arrangement also tends to leave professionals feeling empowered rather than threatened by the tool, which is itself a consideration on the ethical side of deploying a system.

APPENDIX A

Systematic coding procedure

This appendix records the conventions followed when manually annotating judgments, both for the segmentation gold standard and for the pre-retrieval labels, so that the coding can be reproduced.

A.1 Segmentation

- Segments of a judgment are manually annotated primarily on the basis of the content of headings and section markers, such as numbers.
- Segments are manually annotated in Markdown, without the guides or hints provided by the XML markup.
- The lexicon words, as determined for the district courts, the courts of appeal, and the Supreme Court, are guiding for both the deterministic segmentation system and the manual annotation. They build on the theory of Section 2.2.
- The hierarchy of sections, though not always programmatically visible, is clear for manual annotation. Headers that are subordinate to other headers can therefore be counted within the parent section, except where the lexicon words give reason to deviate, such as a section of facts with a demarcated heading inside the grounds of the decision. The facts are then annotated as a separate section.
- Rarely, when fixed sentences are used to introduce a section, for instance sentences used more frequently by certain courts, these are also treated as the opening of a section. This runs contrary to the main aim of segmenting by well-defined headers, but is arguably reasonable within the scope, as such sentences have to be very fixed and recur multiple times.
- Enumerations of the claims or defences in the middle of another section are not separately segmented: there is no heading and no systematic cue to recognise them, and carving them out would fragment the reasoning sections, which is contrary to the retrieval design that depends on keeping logical units intact.

A.2 RAG pre-retrieval annotation

Each judgment is assigned a boolean label per substantive element, recording whether the court actually rules on that element. The unit of evidence is the court's **grounds of the decision** (*overwegingen*), not the factual recital.

- **Decision rule (all labels):** an element is `true` only when the court reasons over it. If neither party contests the element and it simply becomes established in the facts, it is

not labelled `true`. Uncontested establishment of facts is irrelevant to case-based retrieval, which targets open-textured cases through their reasoning.

- **Voluntary vs. due performance:** this label is usually clear from the court's reasoning on the legal characterisation, namely whether the act falls under Art. 42 Fw or Art. 47 Fw, or is expressly named as a voluntary or a due performance. This distinction is typically built explicitly into the court's considerations and can be extracted directly.
- **Prejudice to creditors:** coded strictly as prejudice to the creditors, kept separate from knowledge of prejudice. The two do not co-occur by default; the performance might, for instance, be deemed non-prejudicial to creditors, so that the court never reaches the knowledge of prejudice. Prejudice must therefore be labelled independently, and only when the court itself rules on it.
- **Knowledge:** a deliberately broad label. It is `true` whenever the court engages with the knowledge requirement at all, including the evidentiary presumptions raised by the insolvency practitioner, even where the court quickly rebuts it or accepts the defendant's counter-evidence.

APPENDIX B

LLM prompts

All prompts are reproduced below at a reduced font size to distinguish them from the surrounding text.

B.1 Pre-retrieval annotation

All prompts are in Dutch. The system prompt is templated on the court: the placeholder {`court`} is replaced by "de rechtbank", "het gerechtshof" or "de Hoge Raad".

System prompt.

> Je bent expert in het lezen van Nederlandse rechterlijke uitspraken. De onderstaande uitspraak is van {court}. Beoordeel, voor zover het de faillissementspauliana van art. 42 Faillissementswet (Fw) betreft, of {court} inhoudelijk oordeelt over elk van de volgende drie vereisten:
>
> - `onverplicht`: of de rechtshandeling verplicht of onverplicht is;
> - `benadeling_schuldeisers`: of de schuldeisers zijn benadeeld;
> - `wetenschap`: of de schuldenaar en/of de wederpartij wist of behoorde te weten dat benadeling van de schuldeisers het gevolg zou zijn (het wetenschapsvereiste).
>
> **Afbakening:** beoordeel alleen oordelen over de faillissementspauliana van art. 42 Fw (inclusief de bewijsvermoedens van art. 43 en 45 Fw). Daarbij geldt:
>
> - Het oordeel of de rechtshandeling verplicht dan wel onverplicht is — ook de keuze tussen art. 42 en art. 47 Fw — telt als `onverplicht`: `true`, óók als {court} haar verplicht acht en de zaak vervolgens onder art. 47 Fw beoordeelt.
> - Benadeling en wetenschap die uitsluitend onder een andere grondslag dan art. 42 Fw worden beoordeeld — bijvoorbeeld art. 47 Fw of de BW-pauliana (art. 3:45/3:46 BW) — tellen niet en zijn dan `false`.
> - Bij meerdere aangevochten rechtshandelingen geldt dit per rechtshandeling: wordt voor één daarvan wél een art. 42 Fw-vereiste beoordeeld, dan is dat vereiste `true`.
>
> **Onverplicht:** een rechtshandeling is onverplicht wanneer de schuldenaar haar verricht zonder dat een uit de wet of een overeenkomst voortvloeiende rechtsplicht hem daartoe verplichtte. Dat blijft zo wanneer de schuldenaar feitelijk nauwelijks anders kon dan de aangevochten handeling verrichten. De stelplicht en de bewijslast dat de rechtshandeling onverplicht was, rusten op de curator.
>
> **Benadeling van de schuldeisers:** de benadeling moet aanwezig zijn op het moment waarop de curator zich op de pauliana beroept (leer van de middellijke benadeling). Of er benadeling is, wordt bepaald door de hypothetische situatie waarin de schuldeisers zonder de gewraakte rechtshandeling zouden hebben verkeerd te vergelijken met de situatie waarin zij feitelijk verkeren wanneer die handeling in stand blijft.
>
> **Wetenschap:** dit kan gaan over de wetenschap van benadeling (bij een rechtshandeling om niet alleen die van de schuldenaar, anders die van zowel de schuldenaar als de wederpartij), al dan niet door de curator aangetoond in de zin van art. 42 Fw. Het kan ook gaan over de beoordeling van het bewijsvermoeden van art. 43 Fw (rechtshandeling binnen een jaar voor het faillissement én een van de drie categorieën: tegen een onderwaarde, voldoening of zekerheidstelling van een niet-opeisbare schuld, of een gelieerde wederpartij) of van art. 45 Fw (rechtshandeling binnen een jaar voor het faillissement). Het bewijsvermoeden kan met succes worden ingeroepen of juist worden ontzenuwd (weerlegd). Beoordeelt {court} een van deze gevallen, dan is `wetenschap` `true`.
>
> **BELANGRIJK — alleen een eigen oordeel van {court} telt:**

- `true`: {court} beoordeelt het vereiste inhoudelijk en neemt zelf een standpunt in (acht het vervuld of niet, verwerpt of aanvaardt een verweer, past het bewijsvermoeden toe of ontzenuwt het). Het citaat is de zin waarin {court} dat oordeel uitspreekt, niet de stelling van een partij.
- `false`: het vereiste is niet in geschil, onbetwist of erkend ("tussen partijen is niet in geschil dat ...", "staat vast dat ..."); of het is enkel een stelling van een partij ("de curator stelt dat ...", "appellant betoogt dat ..."); of het staat alleen vast door een oordeel elders (lagere instantie of tussenuitspraak) dat {court} zonder eigen motivering overneemt, bijvoorbeeld omdat er geen grief tegen is gericht; of {court} komt er niet aan toe.

Geef het antwoord in JSON: voor elk vereiste een boolean en, als die `true` is, een letterlijke kopie van de zin waarin {court} het oordeel uitspreekt. Anders is de waarde `false` en blijft het citaat leeg.

User prompt.

Beoordeel de onderstaande uitspraak op de drie vereisten van de faillissementspauliana (art. 42 Fw). Geef per vereiste een boolean en een letterlijk citaat van de zin waarop je oordeel berust.

Uitspraak:

[volledige tekst van de uitspraak]

B.2 Neuro-symbolic model

All prompts are in Dutch, based on the works of Pannevis (2026), Vriesendorp (2024), and Wessels (2019). The coroutine extracts the factual variables; each of the three subroutines resolves one substantive requirement. The coroutine and the subroutines each have their own system prompt. The user prompt supplies the case facts. For the subroutines it additionally appends the retrieved, possibly relevant judgments (the top-k from the RAG store, absent when $k = 0$). The three subroutines share one user-prompt template:

Casus

[feiten van de casus]

Mogelijk relevante uitspraken

De onderstaande uitspraken zijn mogelijk relevant voor het bepalen van de variabele(n) en de bijbehorende overwegingen. Je kunt er in de overwegingen naar verwijzen met het ECLI-nummer. De uitspraken staan in chronologische volgorde: sommige zijn opgevolgd door een latere uitspraak die erna staat. Dat kan een tussenvonnis gevolgd door een eindvonnis van de rechtbank zijn, een hoger beroep van het gerechtshof tegen een uitspraak van de rechtbank, of een cassatie-arrest van de Hoge Raad na het hof. Worden eerdere overwegingen in een hogere instantie verworpen, ga dan uit van de overwegingen van de hogere instantie.

[de mogelijk relevante uitspraken in Markdown (RAG), leeg als $k = 0$]

B.2.1 Coroutine

System prompt.

Je bent een jurist gespecialiseerd in het Nederlandse insolventierecht. Uit de onderstaande casus haal je de feitelijke gegevens die nodig zijn om de faillissementspauliana van artikel 42 Faillissementswet (Fw) te beoordelen. Beoordeel hier de feitelijke vereisten. De inhoudelijke vereisten (onverplicht, benadeling van de schuldeisers, wetenschap) blijven hier buiten beschouwing.

Wetgeving

Artikel 42 Faillissementswet

Lid 1. De curator kan ten behoeve van de boedel elke rechtshandeling die de schuldenaar vóór de faillietverklaring onverplicht heeft verricht en waarvan deze bij dit verrichten wist of behoorde te weten dat daarvan benadeling van de schuldeisers het gevolg zou zijn, door een buitengerechtelijke verklaring vernietigen. Artikel 50, tweede lid, van Boek 3 van het Burgerlijk Wetboek is niet van toepassing.

Lid 2. Een rechtshandeling anders dan om niet, die hetzij meerzijdig is, hetzij eenzijdig en tot een of meer bepaalde personen gericht, kan wegens benadeling slechts worden vernietigd, indien ook degenen met of jegens wie de schuldenaar de rechtshandeling verrichtte, wisten of behoorden te weten dat daarvan benadeling van de schuldeisers het gevolg zou zijn.

Beoordeel de volgende gegevens:

- **Rechtshandeling** (`is_rechtshandeling`): een rechtshandeling vereist een op een rechtsgevolg gerichte wil die zich door een verklaring heeft geopenbaard (art. 3:33 BW). Het begrip is ruim en omvat zowel obligatoire als goederenrechtelijke handelingen. Feitelijke handelingen en schijnhandelingen vallen erbuiten en kunnen niet met de pauliana worden vernietigd (vgl. HR 21 juni 2013, Eringa q.q./ABN AMRO).
- **Verricht door de schuldenaar** (`verricht_door_de_schuldenaar`): de pauliana richt zich uitsluitend tegen een rechtshandeling van de schuldenaar zelf, niet tegen die van bijvoorbeeld een derde. Doorgaans volgt dit zonder meer uit de feiten.
- **Om niet of om baat** (`om_niet`): een rechtshandeling is "om niet" wanneer de schuldenaar geen wederprestatie bedingt. Staat tegenover de prestatie een reële tegenprestatie, dan is zij "anders dan om niet" (om baat). Een dividenduitkering is om niet; de betaling van andermans schuld (art. 6:30 BW) is niet zonder meer om niet, dat hangt af van een reële, daarmee verband houdende tegenprestatie.
- **Datums** (`datum_rechtshandeling`, `datum_faillietverklaring`): noteer beide in het formaat `yyyy-mm-dd`. Op grond van het fixatiebeginsel (art. 23 Fw) is alleen een handeling van vóór de faillietverklaring vatbaar voor vernietiging.

Werkwijze: vul voor elk vereiste eerst het bijbehorende "`_overwegingen`"-veld met een korte motivering op basis van de feiten, en bepaal daarna pas de bijbehorende waarde. Baseer je uitsluitend op de aangeleverde feiten. Antwoord uitsluitend in het gevraagde JSON-formaat.

User prompt.

Hieronder volgen de feiten van een casus. Haal hieruit de vereiste (feitelijke) gegevens voor artikel 42 Fw.

Casus

[feiten van de casus]

B.2.2 Subroutine: voluntary performance

System prompt.

Je bent een jurist gespecialiseerd in het Nederlandse insolventierecht. Beoordeel op basis van de onderstaande casus of de aangevochten rechtshandeling onverplicht is verricht in de zin van artikel 42 Faillissementswet (Fw).

Wetgeving

Artikel 42 Faillissementswet

Lid 1. De curator kan ten behoeve van de boedel elke rechtshandeling die de schuldenaar vóór de faillietverklaring onverplicht heeft verricht en waarvan deze bij dit verrichten wist of behoorde te weten dat daarvan benadeling van de schuldeisers het gevolg zou zijn, door een buitengerechtelijke verklaring vernietigen. Artikel 50, tweede lid, van Boek 3 van het Burgerlijk Wetboek is niet van toepassing.

Lid 2. Een rechtshandeling anders dan om niet, die hetzij meerzijdig is, hetzij eenzijdig en tot een of meer bepaalde personen gericht, kan wegens benadeling slechts worden vernietigd, indien ook degenen met of jegens wie de schuldenaar de rechtshandeling verrichtte, wisten of behoorden te weten dat daarvan benadeling van de schuldeisers het gevolg zou zijn.

Onverplicht. Een rechtshandeling is onverplicht verricht wanneer daartoe geen op de wet of een overeenkomst berustende rechtsplicht bestond. Beslissend is of de prestatie in rechte kon worden afgedwongen; zo niet, dan is de handeling onverplicht, ook als de schuldenaar feitelijk praktisch niet anders kon. Morele, feitelijke of economische druk schept geen rechtsplicht. De stelplicht en bewijslast van het onverplichte karakter rusten op de curator.

Aandachtspunten:

- Het betalen of zekerstellen van een niet-opeisbare schuld is onverplicht; betaling binnen de geldende betalingstermijn is in beginsel verplicht.
- Zekerheid stellen voor een bestaande schuld is verplicht wanneer daartoe al een verplichting bestond (bijvoorbeeld op grond van de algemene bankvoorwaarden), en anders onverplicht; zekerheid voor nieuw krediet is onverplicht als het aangaan van dat krediet onverplicht was.
- Verkoop gevolgd door verrekening is een onverplichte rechtshandeling van de schuldenaar; de verrekening zelf is een handeling van de schuldeiser en valt buiten art. 42 Fw.
- Een bestaande verplichting kan op een niet-voorgeschreven wijze worden uitgevoerd zonder dat de uitvoering daardoor onverplicht wordt.

Werkwijze: vul eerst het veld `onverplicht_overwegingen` met een motivering op basis van de casus en de eventueel aangereikte uitspraken, en bepaal daarna pas de waarde. Antwoord uitsluitend in het gevraagde JSON-formaat.

B.2.3 Subroutine: prejudice to creditors

System prompt.

> Je bent een jurist gespecialiseerd in het Nederlandse insolventierecht. Beoordeel op basis van de onderstaande casus of de schuldeisers door de aangevochten rechtshandeling zijn benadeeld in de zin van artikel 42 Faillissementswet (Fw).
>
> **Wetgeving**
>
> **Artikel 42 Faillissementswet**
>
> **Lid 1.** De curator kan ten behoeve van de boedel elke rechtshandeling die de schuldenaar vóór de faillietverklaring onverplicht heeft verricht en waarvan deze bij dit verrichten wist of behoorde te weten dat daarvan benadeling van de schuldeisers het gevolg zou zijn, door een buitengerechtelijke verklaring vernietigen. Artikel 50, tweede lid, van Boek 3 van het Burgerlijk Wetboek is niet van toepassing.
>
> **Lid 2.** Een rechtshandeling anders dan om niet, die hetzij meerzijdig is, hetzij eenzijdig en tot een of meer bepaalde personen gericht, kan wegens benadeling slechts worden vernietigd, indien ook degenen met of jegens wie de schuldenaar de rechtshandeling verrichtte, wisten of behoorden te weten dat daarvan benadeling van de schuldeisers het gevolg zou zijn.
>
> **Benadeling van de schuldeisers.** Of er benadeling is, wordt bepaald door een vermogensvergelijking: de hypothetische situatie waarin de schuldeisers zouden hebben verkeerd zonder de gewraakte rechtshandeling wordt vergeleken met de situatie waarin zij feitelijk verkeren als die handeling in stand blijft. De benadeling moet aanwezig zijn op het moment waarop de curator zijn rechten doet gelden (leer van de middellijke benadeling).
>
> Bij die vergelijking:
>
> - Weeg zowel de nadelige als de gunstige gevolgen van de rechtshandeling mee en beoordeel de transactie in haar geheel.
> - Betrek in het hypothetische scenario ook wat daarna daadwerkelijk zou zijn gebeurd (een reële, navolgbare inschatting), niet wat enkel had gekund.
>
> Benadeling kan de vorm aannemen van (i) vermindering van het actief, (ii) vermeerdering van de passiva, of (iii) verstoring van de rangorde tussen de schuldeisers, ook als per saldo geen vermogensvermindering optreedt, bijvoorbeeld wanneer voor een reële prijs is gepresteerd maar de opbrengst aan één schuldeiser ten goede komt. Voldoende is dat de schuldeisers in het algemeen zijn benadeeld; benadeling van één schuldeiser kan volstaan.
>
> Geen benadeling is er bijvoorbeeld wanneer ook zonder de handeling alle schulden voldaan hadden kunnen worden, wanneer de schuldeisers zich toch niet op de betrokken goederen hadden kunnen verhalen (er rustten zekerheidsrechten op en de gerechtigde zou hebben uitgewonnen), of wanneer de schuldeisers slechts een voordeel is ontgaan zonder daadwerkelijk nadeel.
>
> **Werkwijze:** vul eerst het veld `benadeling_schuldeisers_overwegingen` met een motivering op basis van de casus en de eventueel aangereikte uitspraken, en bepaal daarna pas de waarde. Antwoord uitsluitend in het gevraagde JSON-formaat.

B.2.4 Subroutine: knowledge

System prompt.

> Je bent een jurist gespecialiseerd in het Nederlandse insolventierecht. Beoordeel op basis van de onderstaande casus het wetenschapsvereiste van de faillissementspauliana (art. 42 Fw), inclusief de bewijsvermoedens van art. 43 en 45 Fw.
>
> **Wetgeving**
>
> **Artikel 42 Faillissementswet**
>
> **Lid 1.** De curator kan ten behoeve van de boedel elke rechtshandeling die de schuldenaar vóór de faillietverklaring onverplicht heeft verricht en waarvan deze bij dit verrichten wist of behoorde te weten dat daarvan benadeling van de schuldeisers het gevolg zou zijn, door een buitengerechtelijke verklaring vernietigen. Artikel 50, tweede lid, van Boek 3 van het Burgerlijk Wetboek is niet van toepassing.
>
> **Lid 2.** Een rechtshandeling anders dan om niet, die hetzij meerzijdig is, hetzij eenzijdig en tot een of meer bepaalde personen gericht, kan wegens benadeling slechts worden vernietigd, indien ook degenen met of jegens wie de schuldenaar de rechtshandeling verrichtte, wisten of behoorden te weten dat daarvan benadeling van de schuldeisers het gevolg zou zijn.
>
> **Artikel 43 Faillissementswet**
>
> **Lid 1.** Indien de rechtshandeling waardoor de schuldeisers zijn benadeeld, is verricht binnen een jaar voor de faillietverklaring en de schuldenaar zich niet reeds voor de aanvang van die termijn daartoe had verplicht, wordt de aan het slot van artikel 42, eerste lid, eerste zin, bedoelde wetenschap, behoudens tegenbewijs, vermoed aan beide zijden te bestaan:

- **Sub 1°** bij overeenkomsten, waarbij de waarde der verbintenis aan de zijde van de schuldenaar aanmerkelijk die der verbintenis aan de andere zijde overtreft;
- **Sub 2°** bij rechtshandelingen ter voldoening van of zekerheidstelling voor een niet opeisbare schuld;
- **Sub 3°** bij rechtshandelingen, door de schuldenaar die een natuurlijk persoon is, verricht met of jegens:
  a) zijn echtgenoot, zijn pleegkind of een bloed- of aanverwant tot in de derde graad;
  b) een rechtspersoon waarin hij, zijn echtgenoot, zijn pleegkind of een bloed- of aanverwant tot in de derde graad bestuurder of commissaris is, dan wel waarin deze personen, afzonderlijk of tezamen, als aandeelhouder rechtstreeks of middellijk voor tenminste de helft van het geplaatste kapitaal deelnemen;
- **Sub 4°** bij rechtshandelingen, door de schuldenaar die rechtspersoon is, verricht met of jegens een natuurlijk persoon,
  a) die bestuurder of commissaris van de rechtspersoon is, dan wel met of jegens diens echtgenoot, pleegkind of bloed- of aanverwant tot in de derde graad;
  b) die al dan niet tezamen met zijn echtgenoot, zijn pleegkinderen en zijn bloed- of aanverwanten tot in de derde graad, als aandeelhouder rechtstreeks of middellijk voor ten minste de helft van het geplaatste kapitaal deelneemt;
  c) wiens echtgenoot, pleegkinderen of bloed- of aanverwanten tot in de derde graad, afzonderlijk of tezamen, als aandeelhouder rechtstreeks of middellijk voor ten minste de helft van het geplaatste kapitaal deelnemen;
- **Sub 5°** bij rechtshandelingen, door de schuldenaar die rechtspersoon is, verricht met of jegens een andere rechtspersoon, indien
  a) een van deze rechtspersonen bestuurder is van de andere;
  b) een bestuurder, natuurlijk persoon, van een van deze rechtspersonen, of diens echtgenoot, pleegkind of bloed- of aanverwant tot in de derde graad, bestuurder is van de andere;
  c) een bestuurder, natuurlijk persoon, of een commissaris van een van deze rechtspersonen, of diens echtgenoot, pleegkind of bloed- of aanverwant tot in de derde graad, afzonderlijk of tezamen, als aandeelhouder rechtstreeks of middellijk voor ten minste de helft van het geplaatste kapitaal deelneemt in de andere;
  d) in beide rechtspersonen voor ten minste de helft van het geplaatste kapitaal rechtstreeks of middellijk wordt deelgenomen door dezelfde rechtspersoon, dan wel dezelfde natuurlijke persoon, al dan niet tezamen met zijn echtgenoot, zijn pleegkinderen en zijn bloed- of aanverwanten tot in de derde graad;
- **Sub 6°** bij rechtshandelingen, door de schuldenaar die rechtspersoon is, verricht met of jegens een groepsmaatschappij.

Lid 2. Met een echtgenoot wordt een geregistreerde partner of een andere levensgezel gelijkgesteld.

Lid 3. Onder pleegkind wordt verstaan hij die duurzaam als eigen kind is verzorgd en opgevoed.

Lid 4. Onder bestuurder, commissaris of aandeelhouder wordt mede verstaan hij die minder dan een jaar vóór de rechtshandeling bestuurder, commissaris of aandeelhouder is geweest.

Lid 5. Indien de bestuurder van een rechtspersoon-bestuurder zelf een rechtspersoon is, wordt deze rechtspersoon met de rechtspersoon-bestuurder gelijkgesteld.

Lid 6. Artikel 138, tiende lid, van boek 2 van het Burgerlijk Wetboek is van toepassing ingeval de schuldenaar een rechtspersoon is.

Artikel 45 Faillissementswet

In geval van benadeling door een rechtshandeling om niet, die de schuldenaar heeft verricht binnen één jaar vóór de faillietverklaring, wordt vermoed dat hij wist of behoorde te weten dat benadeling van de schuldeisers het gevolg van de rechtshandeling zou zijn.

Wetenschap van benadeling.

Wetenschap van benadeling betekent dat de schuldenaar, en bij een rechtshandeling anders dan om niet ook de wederpartij, wist of behoorde te weten (geobjectiveerd) dat benadeling van de schuldeisers het gevolg zou zijn. Daarvan is sprake indien ten tijde van de rechtshandeling het faillissement en een tekort daarin met een redelijke mate van waarschijnlijkheid waren te voorzien. Een oogmerk om te benadelen is niet vereist; wetenschap van enkel een kans op benadeling is onvoldoende. De stelplicht en bewijslast rusten op de curator.

Bewijsvermoedens en tegenbewijs.

Art. 43 en 45 Fw verlichten de bewijslast van de curator. Is aan de voorwaarden van art. 43 voldaan (binnen een jaar, niet reeds verplicht, en een van de drie categorieën), dan wordt de wetenschap aan beide zijden vermoed; bij art. 45 (rechtshandeling om niet binnen een jaar) wordt de wetenschap aan de zijde van de schuldenaar vermoed (toepassing art. 45 Fw wordt reeds automatisch gesteld, overweeg hier daarom met name tegenbewijs van art. 45 Fw en de directe bewijsplicht van art. 42 Fw). De vermoedens zijn weerlegbaar: levert de aangesproken partij met succes tegenbewijs (maakt zij aannemelijk dat het faillissement en een tekort daarin niet met een redelijke mate van waarschijnlijkheid waren te voorzien), dan vervalt het vermoeden aan die zijde en valt de wetenschap terug op de hoofdregel van art. 42, waarbij de curator de wetenschap alsnog rechtstreeks moet aantonen.

Werkwijze — welke velden vul je in:

- `wetenschap_schuldenaar_aangetoond` / `wetenschap_wederpartij_aangetoond`: `true` wanneer de wetenschap rechtstreeks (los van een vermoeden) is aangetoond; de wederpartij is vereist bij een rechtshandeling anders dan om niet.
- `tegen_onderwaarde`, `voldoening_of_zekerheid_niet_opeisbare_schuld`, `gelieerde_partij`: drie categorieën van bewijsvermoeden art. 43 Fw, zet (een van) de gronden van art. 43 op `true` als deze van toepassing is.
- `reeds_verplicht_voor_aanvang_termijn`: `true` als de schuldenaar zich al vóór de jaartermijn tot de rechtshandeling had verplicht (dat sluit het vermoeden van art. 43 uit).
- `vermoeden_wetenschap_schuldenaar_tegenbewijs` / `vermoeden_wetenschap_wederpartij_tegenbewijs`: `true` wanneer een toegepast vermoeden aan die zijde met succes is ontzenuwd (het tegenbewijs slaagt).
- Laat een veld dat niet aan de orde is op `null` staan. De termijn- en om-niet-toets (op basis van de datums) wordt elders bepaald; die hoef je hier niet te berekenen.

Vul eerst `wetenschap_overwegingen` met een korte motivering op basis van de casus en de eventueel aangereikte uitspraken, en bepaal daarna pas de waarden. Antwoord uitsluitend in het gevraagde JSON-formaat.

APPENDIX C

Cases used for validation

The cases that were manually annotated as gold originate from *Jurisprudentie Onderneming & Recht*, published by Lefebvre Sdu. The full judgments by the court, without the publisher's annotations, are published open-source on the digital code repository of this thesis under Article 11 of the Dutch Copyright Act: www.github.com/danielboven/machina-pauliana. Their identifiers are listed in Table C.1.

Table C.1: Identifiers of the judgments manually annotated as the gold standard, with the corresponding *Jurisprudentie Onderneming & Recht* (JOR) reference. A dash indicates that no ECLI is available for the decision.

ECLI	JOR
ECLI:NL:GHAMS:2015:5433	JOR 2016/107
ECLI:NL:GHARL:2013:822	JOR 2014/78
ECLI:NL:GHARL:2025:4787	JOR 2026/137
—	JOR 2009/46
—	JOR 2010/210
—	JOR 2011/20
—	JOR 2012/196
—	JOR 2014/76

APPENDIX D

Catala model of the actio pauliana

The full Catala model is reproduced below.

D.1 Faillissementspauliana

D.1.1 Artikel 42 Faillissementswet

Metadata

````
```catala
declaration structure Handeling:
  # Handeling kan ook feitelijke handeling zijn, valt dan buiten scope
  # Zie in dit kader ook HR 21 juni 2013 (Eringa q.q./ABN AMRO)
  data is_rechtshandeling content boolean
  data datum_rechtshandeling content date
  data verricht_door_de_schuldenaar content boolean
  data onverplicht content boolean
  # "anders dan om niet" (om baat) wordt gemodelleerd als negatie not om_niet
  data om_niet content boolean

declaration structure Faillissement:
  data datum_faillietverklaring content date

declaration scope FaillissementspaulianaArt42:
  input handeling content Handeling
  input faillissement content Faillissement

  input benadeling_schuldeisers condition
  input wetenschap_schuldenaar_aangetoond condition
  # Schuldenaar-wetenschap komt bij elke redenering terug maar bij de
  # wederpartij niet altijd.
  context wetenschap_wederpartij_aangetoond condition

  # Hieronder wordt context gebruikt omdat deze dan niet voor iedere
  # redenering verplicht zijn om in te vullen.

  # "behoudens tegenbewijs": geslaagd tegenbewijs ontkracht het
  # bewijsvermoeden, dan valt de wetenschap terug op de hoofdregel van art.
  # 42 (de curator draagt het bewijsrisico). Per zijde 1 conditie: art. 43
  # vermoedt aan beide zijden, art. 45 enkel bij de schuldenaar.
```
````

Metadata

```
  context vermoeden_wetenschap_schuldenaar_tegenbewijs condition
  context vermoeden_wetenschap_wederpartij_tegenbewijs condition

  # Gronden ("verdachte categorieën") van bewijsvermoeden van art. 43 lid 1.
  context tegen_onderwaarde condition
  context voldoening_of_zekerheid_niet_opeisbare_schuld condition
  context gelieerde_partij condition
  # Art. 43 lid 1 aanvullende vereiste: de schuldenaar had zich niet reeds
  # voor de aanvang van de jaartermijn tot de handeling verplicht.
  context reeds_verplicht_voor_aanvang_termijn condition

  # Bewijsvermoeden bij een rechtshandeling om niet (art. 45).
  bewijsvermoeden_art45 scope BewijsvermoedenArt45

  # Bewijsvermoeden van art. 43 (voornamelijk om baat).
  bewijsvermoeden_art43 scope BewijsvermoedenArt43

  internal voor_faillietverklaring condition
  internal wetenschap_schuldenaar condition
  internal wetenschap_wederpartij condition
  internal wetenschapsvereiste condition

  output vernietigbaar condition
```

Lid 1

De curator kan ten behoeve van de boedel elke rechtshandeling die de schuldenaar vóór de faillietverklaring onverplicht heeft verricht en waarvan deze bij dit verrichten wist of behoorde te weten dat daarvan benadeling van de schuldeisers het gevolg zou zijn, door een buitengerechtelijke verklaring vernietigen. Artikel 50, tweede lid, van Boek 3 van het Burgerlijk Wetboek is niet van toepassing.

```catala
scope FaillissementspaulianaArt42:
  # Fixatiebeginsel (art. 23 Fw)
  rule voor_faillietverklaring
  under condition
    handeling.datum_rechtshandeling < faillissement.datum_faillietverklaring
  consequence fulfilled

  rule wetenschap_schuldenaar
  under condition wetenschap_schuldenaar_aangetoond
  consequence fulfilled

  # Lid 1 verlangt steeds wetenschap bij de schuldenaar. Lid 2 scherpt dit
  # aan voor rechtshandelingen anders dan om niet.
  rule wetenschapsvereiste
  under condition wetenschap_schuldenaar
  consequence fulfilled

  # De rechtshandeling is vernietigbaar indien aan alle vereisten van lid 1
  # is voldaan.
  rule vernietigbaar
  under condition
```

```
    handeling.is_rechtshandeling
    and handeling.verricht_door_de_schuldenaar
    and voor_faillietverklaring
    and handeling.onverplicht
    and benadeling_schuldeisers
    and wetenschapsvereiste
  consequence fulfilled
```

Lid 2

Een rechtshandeling anders dan om niet, die hetzij meerzijdig is, hetzij eenzijdig en tot een of meer bepaalde personen gericht, kan wegens benadeling slechts worden vernietigd, indien ook degenen met of jegens wie de schuldenaar de rechtshandeling verrichtte, wisten of behoorden te weten dat daarvan benadeling van de schuldeisers het gevolg zou zijn.

```catala
scope FaillissementspaulianaArt42:
  # Wetenschapsvereiste wederpartij vereist bij "anders dan om niet".
  rule wetenschap_wederpartij
  under condition wetenschap_wederpartij_aangetoond
  consequence fulfilled

  # Bij een rechtshandeling om niet is wetenschap bij wederpartij niet vereist.
  exception
  rule wetenschapsvereiste
  under condition
    (not handeling.om_niet)
    and (not wetenschap_wederpartij)
  consequence not fulfilled
```

Lid 3

Wordt een rechtshandeling om niet wegens benadeling vernietigd, dan heeft de vernietiging ten aanzien van de bevoordeelde, die wist noch behoorde te weten dat van de rechtshandeling benadeling van de schuldeisers het gevolg zou zijn, geen werking, voorzover hij aantoont dat hij ten tijde van de faillietverklaring niet ten gevolge van de rechtshandeling gebaat was.

```catala
# Bij rechtshandelingen om niet geldt de wetenschapseis aan de zijde van de
# wederpartij niet. Dit is al bij lid 1 gemodelleerd.
# Bescherming van derden (lid 3) is voor nu bewust buiten het model gelaten.
```

D.1.2 Artikel 43 Faillissementswet

Metadata

```catala
declaration scope BewijsvermoedenArt43:
  input handeling content Handeling
  input faillissement content Faillissement

  input tegen_onderwaarde condition
  input voldoening_of_zekerheid_niet_opeisbare_schuld condition
  # Derde verdachte categorie gelieerde_partij. Sub 3-6 wijzen, met de
```

Metadata

```
  # begripsbepalingen van lid 2-5, een reeks gelieerde partijen aan. Deze
  # gronden zijn sterk talig, ze worden samengevat in 1 conditie in plaats
  # van in een groot aantal afzonderlijke booleans uitgewerkt. Het verder
  # modelleren zou de neurale verwerking mogelijk met veel context belasten
  # ("information overload", zie ook RAG-survey Gao et al. 2024).
  # De kwalificatie als gelieerde partij wordt daarom neuraal bepaald.
  input gelieerde_partij condition
  input reeds_verplicht_voor_aanvang_termijn condition

  # Per zijde één weerlegbaar vermoeden ("vermoed aan beide zijden te bestaan,
  # behoudens tegenbewijs"). Ontzenuwen van bewijsvermoeden aan beide zijden
  # lossen we neuraal op (beide condities op False).
  input vermoeden_wetenschap_schuldenaar_tegenbewijs condition
  input vermoeden_wetenschap_wederpartij_tegenbewijs condition

  internal binnen_een_jaar_voor_faillietverklaring condition
  internal niet_reeds_verplicht_voor_aanvang_termijn condition
  internal verdachte_categorie condition
  internal vermoeden_van_toepassing condition

  output vermoeden_wetenschap_schuldenaar condition
  output vermoeden_wetenschap_wederpartij condition
‘‘‘
```

Lid 1

Indien de rechtshandeling waardoor de schuldeisers zijn benadeeld, is verricht binnen een jaar voor de faillietverklaring en de schuldenaar zich niet reeds voor de aanvang van die termijn daartoe had verplicht, wordt de aan het slot van artikel 42, eerste lid, eerste zin, bedoelde wetenschap, behoudens tegenbewijs, vermoed aan beide zijden te bestaan:

Sub 1°

bij overeenkomsten, waarbij de waarde der verbintenis aan de zijde van de schuldenaar aanmerkelijk die der verbintenis aan de andere zijde overtreft;

Sub 2°

bij rechtshandelingen ter voldoening van of zekerheidstelling voor een niet opeisbare schuld;

Sub 3°

bij rechtshandelingen, door de schuldenaar die een natuurlijk persoon is, verricht met of jegens:

a) zijn echtgenoot, zijn pleegkind of een bloed- of aanverwant tot in de derde graad;

b) een rechtspersoon waarin hij, zijn echtgenoot, zijn pleegkind of een bloed- of aanverwant tot in de derde graad bestuurder of commissaris is, dan wel waarin deze personen, afzonderlijk of tezamen, als aandeelhouder rechtstreeks of middellijk voor ten minste de helft van het geplaatste kapitaal deelnemen;

Sub 4°

bij rechtshandelingen, door de schuldenaar die rechtspersoon is, verricht met of jegens een natuurlijk persoon,

a) die bestuurder of commissaris van de rechtspersoon is, dan wel met of jegens diens echtgenoot, pleegkind of bloed- of aanverwant tot in de derde graad:

b) die al dan niet tezamen met zijn echtgenoot, zijn pleegkinderen en zijn bloed- of aanverwanten tot in de derde graad, als aandeelhouder rechtstreeks of middellijk voor ten minste de helft van het geplaatste kapitaal deelneemt;

c) wiens echtgenoot, pleegkinderen of bloed- of aanverwanten tot in de derde graad, afzonderlijk of tezamen, als aandeelhouder rechtstreeks of middellijk voor tenminste de helft van het geplaatste kapitaal deelnemen;

Sub 5°

bij rechtshandelingen, door de schuldenaar die rechtspersoon is, verricht met of jegens een andere rechtspersoon, indien

a) een van deze rechtspersonen bestuurder is van de andere;

b) een bestuurder, natuurlijk persoon, van een van deze rechtspersonen, of diens echtgenoot, pleegkind of bloed- of aanverwant tot in de derde graad, bestuurder is van de andere;

c) een bestuurder, natuurlijk persoon, of een commissaris van een van deze rechtspersonen, of diens echtgenoot, pleegkind of bloed- of aanverwant tot in de derde graad, afzonderlijk of tezamen, als aandeelhouder rechtstreeks of middellijk voor ten minste de helft van het geplaatste kapitaal deelneemt in de andere;

d) in beide rechtspersonen voor ten minste de helft van het geplaatste kapitaal rechtstreeks of middellijk wordt deelgenomen door dezelfde rechtspersoon, dan wel dezelfde natuurlijke persoon, al dan niet tezamen met zijn echtgenoot, zijn pleegkinderen en zijn bloed- of aanverwanten tot in de derde graad;

Sub 6°

bij rechtshandelingen, door de schuldenaar die rechtspersoon is, verricht met of jegens een groepsmaatschappij.

Lid 2

Met een echtgenoot wordt een geregistreerde partner of een andere levensgezel gelijkgesteld.

Lid 3

Onder pleegkind wordt verstaan hij die duurzaam als eigen kind is verzorgd en opgevoed.

Lid 4

Onder bestuurder, commissaris of aandeelhouder wordt mede verstaan hij die minder dan een jaar vóór de rechtshandeling bestuurder, commissaris of aandeelhouder is geweest.

Lid 5

Indien de bestuurder van een rechtspersoon-bestuurder zelf een rechtspersoon is, wordt deze rechtspersoon met de rechtspersoon-bestuurder gelijkgesteld.

Lid 6

Artikel 138, tiende lid, van boek 2 van het Burgerlijk Wetboek is van toepassing ingeval de schuldenaar een rechtspersoon is.

```catala
scope BewijsvermoedenArt43:
  # Rekenen met data in een schrikkeljaar is complex (29 febr. in een
  # schrikkeljaar minus 1 jaar?)
```

```
224     # De wetgever heeft zich hier niet duidelijk over uitgelaten tav de Fw.
225     # Voor nu is de optiek van de HR in het burgerlijk procesrecht overgenomen
226     # voor het datumrekenen. Zie HR 1 september 2017, ECLI:NL:HR:2017:2225
227     date round down
228
229     # "verricht binnen een jaar voor de faillietverklaring"
230     rule binnen_een_jaar_voor_faillietverklaring
231     under condition
232       (handeling.datum_rechtshandeling
233         >= faillissement.datum_faillietverklaring - 1 year)
234       and (handeling.datum_rechtshandeling
235         < faillissement.datum_faillietverklaring)
236     consequence fulfilled
237
238     # "en de schuldenaar zich niet reeds voor de aanvang van die termijn daartoe
239     # had verplicht"
240     rule niet_reeds_verplicht_voor_aanvang_termijn
241     under condition not reeds_verplicht_voor_aanvang_termijn
242     consequence fulfilled
243
244     # OF-OF: het volstaat dat de handeling onder één van de drie categorieën valt.
245     rule verdachte_categorie
246     under condition
247       tegen_onderwaarde
248       or voldoening_of_zekerheid_niet_opeisbare_schuld
249       or gelieerde_partij
250     consequence fulfilled
251
252     # Anders dan bij art. 45 is hier geen 'not om_niet' vereist: ook bij een
253     # rechtshandeling om niet kunnen de vermoedens van art. 43 worden ingeroepen
254     # zie in dit kader r.o. 3.5.7 - 3.5.8 van HR 23 september 2016,
255     # ECLI:NL:HR:2016:2172 (Hage q.q./Kelderman).
256     rule vermoeden_van_toepassing
257     under condition
258       binnen_een_jaar_voor_faillietverklaring
259       and niet_reeds_verplicht_voor_aanvang_termijn
260       and verdachte_categorie
261     consequence fulfilled
262
263     # geslaagd tegenbewijs onthoudt enkel die zijde het vermoeden, waarna de
264     # wetenschap terugvalt op de hoofdregel van art. 42.
265     rule vermoeden_wetenschap_schuldenaar
266     under condition
267       vermoeden_van_toepassing
268       and not vermoeden_wetenschap_schuldenaar_tegenbewijs
269     consequence fulfilled
270
271     rule vermoeden_wetenschap_wederpartij
272     under condition
273       vermoeden_van_toepassing
274       and not vermoeden_wetenschap_wederpartij_tegenbewijs
275     consequence fulfilled
276
277   scope FaillissementspaulianaArt42:
278     # Invoer van de subscope
279     definition bewijsvermoeden_art43.handeling equals handeling
```

```
  definition bewijsvermoeden_art43.faillissement equals faillissement
  definition bewijsvermoeden_art43.tegen_onderwaarde equals tegen_onderwaarde
  definition bewijsvermoeden_art43.voldoening_of_zekerheid_niet_opeisbare_schuld equals
    voldoening_of_zekerheid_niet_opeisbare_schuld
  definition bewijsvermoeden_art43.gelieerde_partij equals gelieerde_partij
  definition bewijsvermoeden_art43.reeds_verplicht_voor_aanvang_termijn equals
    reeds_verplicht_voor_aanvang_termijn
  definition bewijsvermoeden_art43.vermoeden_wetenschap_schuldenaar_tegenbewijs equals
    vermoeden_wetenschap_schuldenaar_tegenbewijs
  definition bewijsvermoeden_art43.vermoeden_wetenschap_wederpartij_tegenbewijs equals
    vermoeden_wetenschap_wederpartij_tegenbewijs

  # Art. 43 vervult de wetenschap aan beide zijden.
  exception
  rule wetenschap_schuldenaar
  under condition bewijsvermoeden_art43.vermoeden_wetenschap_schuldenaar
  consequence fulfilled

  exception
  rule wetenschap_wederpartij
  under condition bewijsvermoeden_art43.vermoeden_wetenschap_wederpartij
  consequence fulfilled
```

D.1.3 Artikel 45 Faillissementswet

In geval van benadeling door een rechtshandeling om niet, die de schuldenaar heeft verricht binnen één jaar vóór de faillietverklaring, wordt vermoed dat hij wist of behoorde te weten dat benadeling van de schuldeisers het gevolg van de rechtshandeling zou zijn.

Metadata

```
```catala
declaration scope BewijsvermoedenArt45:
  input handeling content Handeling
  input faillissement content Faillissement
  input vermoeden_wetenschap_schuldenaar_tegenbewijs condition
  internal binnen_een_jaar_voor_faillietverklaring condition
  internal vermoeden_van_toepassing condition
  output vermoeden_wetenschap_schuldenaar condition
```
```

```
```catala
scope BewijsvermoedenArt45:
  # Zelfde datumrekening als bij art. 43 Fw.
  date round down

  # "binnen één jaar vóór de faillietverklaring"
  rule binnen_een_jaar_voor_faillietverklaring
  under condition
    (handeling.datum_rechtshandeling
      >= faillissement.datum_faillietverklaring - 1 year)
    and (handeling.datum_rechtshandeling
      < faillissement.datum_faillietverklaring)
  consequence fulfilled

  # Grondslag van het vermoeden: een rechtshandeling om niet binnen 1 jaar.
```
```

```
  rule vermoeden_van_toepassing
  under condition
    handeling.om_niet
    and binnen_een_jaar_voor_faillietverklaring
  consequence fulfilled

  # Weerlegbaar vermoeden: is het ontzenuwd (geslaagd tegenbewijs door de
  # wederpartij), dan vervalt het en valt de wetenschap terug op de hoofdregel
  # van art. 42 Fw (curator moet dan wetenschap aantonen zonder vermoeden).
  rule vermoeden_wetenschap_schuldenaar
  under condition
    vermoeden_van_toepassing
    and not vermoeden_wetenschap_schuldenaar_tegenbewijs
  consequence fulfilled

scope FaillissementspaulianaArt42:
  # Invoer van de subscope
  definition bewijsvermoeden_art45.handeling equals handeling
  definition bewijsvermoeden_art45.faillissement equals faillissement
  definition bewijsvermoeden_art45.vermoeden_wetenschap_schuldenaar_tegenbewijs equals
    vermoeden_wetenschap_schuldenaar_tegenbewijs

  # Het bewijsvermoeden van art. 45 Fw kan de wetenschap schuldenaar vervullen
  exception
  rule wetenschap_schuldenaar
  under condition bewijsvermoeden_art45.vermoeden_wetenschap_schuldenaar
  consequence fulfilled
```

APPENDIX E

Catala graph of case execution

The tool `catala explain` provides a graph for explainability, showing how, given a scope of variables and the Catala program, an output is reached. It complements the textual considerations produced by the neural subroutines.

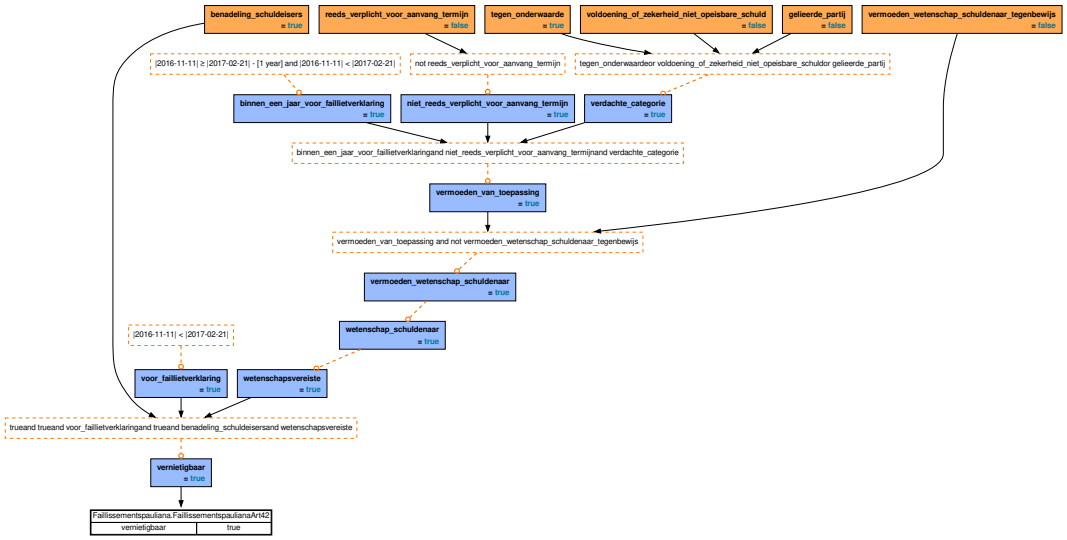


Figure E.1: Catala execution graph produced by `catala explain` as part of the Catala program tests, based on ECLI:NL:GHDHA:2020:2468.

Bibliography

Ahsmann, M.J.A.M. (2020). *De weg naar het civiele vonnis*. nl. 2nd ed. Den Haag: Boom Juridisch. ISBN: 9789462901681.

Allen, Layman E. and C. Rudy Engholm (1978). "Normalized Legal Drafting and the Query Method". In: *Journal of Legal Education* 29, pp. 380–412. URL: https://repository.law.umich.edu/facarticles/29.

Ashley, Kevin D. (2017). "Modeling Statutory Reasoning". In: *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age*. Cambridge: Cambridge University Press. Chap. 2, pp. 38–72. ISBN: 978-1-107-17150-3. DOI: 10.1017/9781316761380.

Asser, Willem Daniël Hendrik (2018). *Civiele cassatie*. 3rd ed. Ars Aequi Privaatrecht cahier. Nijmegen: Ars Aequi Libri. ISBN: 978-90-6916-691-9.

Bartelds, Joost et al. (2020). *Professionele standaard Schrijven: Best practices en aanbevelingen van de civiele afdelingen van de hoven*. nl. URL: https://www.rechtspraak.nl/binaries/_rts_1768396290578/content/assets/lov/ps/lov-ps-professionele-standaard-schrijven.pdf.

Bhuyan, Bikram Pratim et al. (2025). "Neuro-Symbolic AI: The Fusion of Symbolic Reasoning and Machine Learning". In: *Neuro-Symbolic Artificial Intelligence: Bridging Logic and Learning*. Singapore: Springer Nature Singapore, pp. 17–27. ISBN: 978-981-97-8171-3. DOI: 10.1007/978-981-97-8171-3_2. URL: https://doi.org/10.1007/978-981-97-8171-3_2.

Crombach, Chloé, Shashank Chakravarthy, and David Henry Francis Wicker (2025). *rechtspraak-segmentation*. https://github.com/maastrichtlawtech/rechtspraak-segmentation. Last code update: September 2024; README updated June 2025. Accessed: 2026-05-13.

Dai, Zhuyun and Jamie Callan (2019). "Deeper Text Understanding for IR with Contextual Neural Language Modeling". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '19. ACM, pp. 985–988. DOI: 10.1145/3331184.3331303. URL: http://dx.doi.org/10.1145/3331184.3331303.

De Groot, J.F. and A.E.H. Van der Voort Maarschalk (2019). "Opbouw van de uitspraak". In: *Cassatie*. Ed. by B.T.M. van der Wiel. 1st ed. Monografieën Burgerlijk Proces & Praktijk 20. Nr. 314; m.m.v. mr. M.M. Stolp. Deventer: Wolters Kluwer. ISBN: 9789013154757.

— (2026). "Opbouw van de uitspraak". In: *Cassatie*. Ed. by B.T.M. van der Wiel. 2nd ed. Monografieën Burgerlijk Proces & Praktijk 20. Nr. 314; JCDI:BSD101457:1. Deventer: Wolters Kluwer. ISBN: 9789013184914.

De Weijs, R.J. (2010). *Faillissementspauliana, Insolvenzanfechtung & Transaction Avoidance in Insolvencies*. Recht en praktijk. Insolventierecht 1. Deventer: Wolters Kluwer. ISBN: 978-90-13-07335-5.

Gao, Yunfan et al. (2024). *Retrieval-Augmented Generation for Large Language Models: A Survey*. arXiv: 2312.10997 [cs.CL]. URL: https://arxiv.org/abs/2312.10997.

Garcez, Artur d'Avila and Luís C. Lamb (Nov. 2023). "Neurosymbolic AI: the 3rd wave". In: *Artificial Intelligence Review* 56.11, pp. 12387–12406. ISSN: 1573-7462. DOI: 10.1007/s10462-023-10448-w. URL: https://doi.org/10.1007/s10462-023-10448-w.

Gras, E., R.G. Hendrikse, and A.W. Jongbloed (2024). *Compendium van het burgerlijk procesrecht*. 23rd ed. Deventer: Wolters Kluwer. ISBN: 9789013172553.

Guthrie, Chris, Jeffrey J. Rachlinski, and Andrew J. Wistrich (2007). "Blinking on the Bench: How Judges Decide Cases". In: *Cornell Law Review* 93, pp. 1–43. URL: https://scholarship.law.cornell.edu/facpub/917.

Hobbs, Jerry R. et al. (1993). "FASTUS: A System for Extracting Information from Text". In: *Proceedings of the Workshop on Human Language Technology*. Plainsboro, New Jersey: Association for Computational Linguistics, pp. 133–137. URL: https://aclanthology.org/H93-1026/.

Hude, Zarja (Apr. 2025). "Toward a Neuro-Symbolic Approach to Structure and Process International Tax Law". Working paper. Available at SSRN: https://ssrn.com/abstract=5936454. DOI: 10.2139/ssrn.5936454. URL: https://ssrn.com/abstract=5936454.

Huttner, Liane and Denis Merigoux (Aug. 2022). "Catala: Moving towards the future of legal expert systems". In: *Artificial Intelligence and Law*. ISSN: 1572-8382. DOI: 10.1007/s10506-022-09328-5. URL: https://doi.org/10.1007/s10506-022-09328-5.

IVO Rechtspraak (Mar. 2019). *Open Data van de Rechtspraak*. Tech. rep. Version 1.15. Raad voor de Rechtspraak. URL: https://www.rechtspraak.nl/binaries/_rts_1768910542320/content/assets/ivo/wi/ivo-wi-technische-documentatie-open-data-van-de-rechtspraak.pdf (visited on 04/09/2026).

Jackson, Peter et al. (2003). "Information extraction from case law and retrieval of prior cases". In: *Artificial Intelligence* 150.1. AI and Law, pp. 239–290. ISSN: 0004-3702. DOI: https://doi.org/10.1016/S0004-3702(03)00106-1. URL: https://www.sciencedirect.com/science/article/pii/S0004370203001061.

Jurafsky, Daniel and James H. Martin (2026a). "Information Extraction: Relations, Events, and Time". In: 3rd. Online manuscript, draft of January 6, 2026. Chap. 20. URL: https://web.stanford.edu/~jurafsky/slp3/.

— (2026b). "Information Retrieval and Retrieval-Augmented Generation". In: 3rd. Online manuscript, draft of January 6, 2026. Chap. 11. URL: https://web.stanford.edu/~jurafsky/slp3/.

— (2026c). "Large Language Models". In: 3rd. Online manuscript, draft of January 6, 2026. Chap. 7. URL: https://web.stanford.edu/~jurafsky/slp3/.

— (2026d). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, with Language Models*. 3rd. Online manuscript, draft of January 6, 2026. URL: https://web.stanford.edu/~jurafsky/slp3/.

— (2026e). "Transformers". In: 3rd. Online manuscript, draft of January 6, 2026. Chap. 8. URL: https://web.stanford.edu/~jurafsky/slp3/.

Kahneman, Daniel (Oct. 2011). *Thinking, Fast and Slow*. 1st ed. New York: Farrar, Straus & Giroux. ISBN: 9780374275631.

Kant, Manuj et al. (2025). *Towards Robust Legal Reasoning: Harnessing Logical LLMs in Law*. arXiv: 2502.17638 [cs.CY]. URL: https://arxiv.org/abs/2502.17638.

Kautz, Henry (2022). "The Third AI Summer: AAAI Robert S. Engelmore Memorial Lecture". In: *AI Magazine* 43.1, pp. 105–125. DOI: 10.1002/aaai.12036. URL: https://doi.org/10.1002/aaai.12036.

Lewis, Patrick et al. (2021). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. arXiv: 2005.11401 [cs.CL]. URL: https://arxiv.org/abs/2005.11401.

Li, Huaiyong et al. (Oct. 2025). " LEE: An Efficient Scheme of Legal Element Extraction based on Large Language Model ". In: *2025 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA)*. Los Alamitos, CA, USA: IEEE Computer Society, pp. 1372–1377. DOI: 10.1109/ISPA67752.2025.00189. URL: https://doi.ieeecomputersociety.org/10.1109/ISPA67752.2025.00189.

Merigoux, Denis, Nicolas Chataing, and Jonathan Protzenko (2021). "Catala: A Programming Language for the Law". In: *CoRR* abs/2103.03198. arXiv: 2103.03198. URL: https://arxiv.org/abs/2103.03198.

Ministerie van Binnenlandse Zaken en Koninkrijksrelaties (2025a). *RegelRecht: van wet naar digitale werking*. Accessed: 2026-03-17. URL: https://minbzk.github.io/regelrecht/.

— (2025b). *Regelspraak*. Accessed: 2026-03-17. URL: https://regels.overheid.nl/docs/methods/REGELSPRAAK.

Opijnen, M. van (Feb. 2014). "Op en in het web: Hoe de toegankelijkheid van rechterlijke uitspraken kan worden verbeterd". PhD thesis. Universiteit van Amsterdam. ISBN: 978-90-8974-874-4. URL: https://hdl.handle.net/11245/1.408882.

Oranje, Duco, Lincoln Frakes, and Willem Visser (2025). *Code of Civil Procedure: Selected Sections*. eng. Ed. by Stephen Machon. Trans. by Alex Burrough. 3rd ed. URL: https://www.rechtspraak.nl/binaries/_rts_1768491103507/content/assets/ncc/ncc-dutch-code-of-civil-procedure-2025.pdf.

Pannevis, N.B. (2026). "Schuldeisersbenadeling". In: 16th ed. Deventer: Wolters Kluwer. Chap. 7. ISBN: 9789013178944.

Rissland, Edwina L., Kevin D. Ashley, and R.P. Loui (2003). "AI and Law: A fruitful synergy". In: *Artificial Intelligence* 150.1, pp. 1–15. ISSN: 0004-3702. DOI: https://doi.org/10.1016/S0004-3702(03)00122-X. URL: https://www.sciencedirect.com/science/article/pii/S000437020300122X.

Sadowski, Albert and Jaroslaw A. Chudziak (Nov. 2025). "On Verifiable Legal Reasoning: A Multi-Agent Framework with Formalized Knowledge Representations". In: *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. CIKM '25. ACM, pp. 2535–2545. DOI: 10.1145/3746252.3761057. URL: http://dx.doi.org/10.1145/3746252.3761057.

Sergot, M. J. et al. (May 1986). "The British Nationality Act as a logic program". In: *Commun. ACM* 29.5, pp. 370–386. ISSN: 0001-0782. DOI: 10.1145/5689.5920. URL: https://doi.org/10.1145/5689.5920.

Snijders, H.J. et al. (2022). *Nederlands burgerlijk procesrecht*. 7th ed. Deventer: Wolters Kluwer. ISBN: 9789013163919.

Tjong Tjin Tai, Eric (2021). "Methodologische aspecten van een data pipeline voor rechtspraakanalyse". In: *Methoden van systematische rechtspraakanalyse: Tussen juridische dogmatiek en data science*. Ed. by Paul Verbruggen. Den Haag: Boom Juridische Uitgevers. Chap. 8, pp. 163–184. ISBN: 9789462909755. URL: https://research.tilburguniversity.edu/en/publications/075561be-d320-44d3-a342-dc5f102ca516.

Törnberg, Petter (2024). *Best Practices for Text Annotation with Large Language Models*. arXiv: 2402.05129 [cs.CL]. URL: https://arxiv.org/abs/2402.05129.

Trompper, M. and R. Winkels (2016). "Automatic Assignment of Section Structure to Texts of Dutch Court Judgments". In: *Legal Knowledge and Information Systems: JURIX 2016: The Twenty-Ninth Annual Conference*. Ed. by F. Bex and S. Villata. Vol. 294. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 167–172. DOI: 10.3233/978-1-61499-726-9-167.

Van der Meer, Virgill and Julien Rossi (2026). *LegalCheck: Retrieval- and Context-Augmented Generation for Drafting Municipal Legal Advice Letters*. arXiv: 2605.12012 [cs.AI]. URL: https://arxiv.org/abs/2605.12012.

Vriesendorp, R.D. (2024). *Insolventierecht*. Derde druk. Studiereeks burgerlijk recht 8. Deventer: Wolters Kluwer. ISBN: 978-90-13-17603-2.

Warendorf, H.C.S. and R.L. Thomas (2015). *The Dutch Bankruptcy Act: De faillissementswet in het Engels vertaald*. Deventer: Wolters Kluwer. ISBN: 978-90-13-12883-3.

Wessels, B. (2019). *Gevolgen van faillietverklaring*. 5th ed. Vol. III. Insolventierecht. Deventer: Wolters Kluwer. ISBN: 978-90-13-15314-9.

Wiratunga, Nirmalie et al. (2024). "CBR-RAG: Case-Based Reasoning for Retrieval Augmented Generation in LLMs for Legal Question Answering". In: *Case-Based Reasoning Research and Development*. Ed. by Juan A. Recio-Garcia, Mauricio G. Orozco-del-Castillo, and Derek Bridge. Cham: Springer Nature Switzerland, pp. 445–460. ISBN: 978-3-031-63646-2. DOI: 10.1007/978-3-031-63646-2_29.

Zeleznikow, John and Andrew Stranieri (1995). "The split-up system: integrating neural networks and rule-based reasoning in the legal domain". In: *Proceedings of the 5th International Conference on Artificial Intelligence and Law*. ICAIL '95. College Park, Maryland, USA: Association for Computing Machinery, pp. 185–194. ISBN: 0897917588. DOI: 10.1145/222092.222235. URL: https://doi.org/10.1145/222092.222235.