Artificial Intelligence and Law (2022) 30:163–197
https://doi.org/10.1007/s10506-021-09295-3

ORIGINAL RESEARCH

Logical English meets legal English for swaps and derivatives

Robert Kowalski[1] · **Akber Datoo**[2]

Accepted: 24 June 2021 / Published online: 12 August 2021
© The Author(s) 2021

Abstract

In this paper, we present an informal introduction to Logical English (LE) and illustrate its use to standardise the legal wording of the Automatic Early Termination (AET) clauses of International Swaps and Derivatives Association (ISDA) Agreements. LE can be viewed both as an alternative to conventional legal English for expressing legal documents, and as an alternative to conventional computer languages for automating legal documents. LE is a controlled natural language (CNL), which is designed both to be computer-executable and to be readable by English speakers without special training. The basic form of LE is syntactic sugar for logic programs, in which all sentences have the same standard form, either as rules of the form *conclusion if conditions* or as unconditional sentences of the form *conclusion.* However, LE extends normal logic programming by introducing features that are present in other computer languages and other logics. These features include typed variables signalled by common nouns, and existentially quantified variables in the *conclusions* of sentences signalled by indefinite articles. Although LE translates naturally into a logic programming language such as Prolog or ASP, it can also serve as a neutral standard, which can be compiled into other lower-level computer languages.

Keywords Logical English · Controlled Natural Language · Logic Programming · Clause taxonomy · ISDA · Close-out netting

✉ Robert Kowalski
rak@doc.ic.ac.uk

Akber Datoo
akber@d2legaltech.com

[1] Department of Computing, Imperial College London, London, UK

[2] D2 Legal Technology, London, UK

1 Introduction

Logical English (LE) (Kowalski 1982, 1990, 2019, 2020) is a work in progress, intended as syntactic sugar for logic programs, which consist of "atomic sentences", such as:

IsdaAgreement is dated as of 03/10/2020.

and conditional sentences (or rules), such as[1]:

A Transaction is governed by IsdaAgreement
if a confirmation of the Transaction states that
the Transaction is governed by IsdaAgreement
and IsdaAgreement is dated as of a time T1
and the Transaction commences at a time T2
and T1 ≤ T2.

Here the most significant logical keywords have been highlighted in bold. In general, rules in LE have the form:

conclusion ***if*** *conditions*.

where the *conclusion* is an atomic sentence and the *conditions* are a conjunction of atomic sentences or their negations.

LE is intended for use as a Turing-complete general-purpose computer language, covering a spectrum of programming, database and AI knowledge representation applications. However, recent prototypes of LE have focussed primarily on legal applications (Davila, 2017; Karadotchev, 2019; Fu, 2020). The guiding principles of its design are that it be:

- understandable without training in computing, logic or advanced mathematics;
- efficiently executable; and
- as unambiguous as possible, to reduce human misunderstanding, and to facilitate computer executability.

In this paper we present both an introduction to LE and an application of LE to the logical analysis and representation of legal clauses concerning Automatic Early Termination (AET) of International Swaps and Derivatives Association (ISDA) Master Agreements. The paper has been written both for those who might want to explore LE as an alternative to conventional legal English for expressing legal documents, and for those who want to explore LE as an alternative to conventional computer languages for automating legal documents.

[1] The use of capitals here has no significance, but reflects the convention in ISDA agreements "that terms with initial capitals have the meanings attributed to those terms in the Agreement".

ISDA Master Agreements are complex legal documents for governing over-the-counter (OTC) derivatives transactions. But despite their complexity, it is possible to analyse their logic without having a deep understanding of their financial markets and legal content, similar to the way in which it is possible for a computer to execute a program without the computer understanding the meaning of the program. It is with this initial and limited understanding of ISDA Master Agreements that the first author undertook the first phase of the work reported in this paper. In the second phase, the second author helped to refine the logical analysis and drew attention to errors, taking the financial and legal context into account.

This paper does not assume any previous background in formal logic, computing, finance or law. Although some readers may have difficulty with some of the topics falling outside their own area of expertise (as did the authors), they should be able to discern the general drift of the paper, and to judge its applicability to their own interests.

It is important that the reader does not confuse any difficulty they might have with reading this paper with the difficulty a person might have with reading LE itself, because a person reading LE would not need the detailed explanations of some of the alternative LE formulations discussed in this paper. It is also important that the reader does not confuse any difficulty they might have with reading this paper with the difficulty a person would have with writing LE. We will discuss the challenges involved in writing LE in the related work and conclusion sections of the paper.

In the remainder of the paper, we present an overview of ISDA Master Agreements, an overview of LE and its treatment of rules and exceptions, the LE representation of Section 6(a) of the ISDA Master Agreement, which deals with Automatic Early Termination, and the LE representation of a sample of AET Schedule clauses, followed by a discussion of related work, and some final thoughts in the Conclusion.

2 ISDA Master Agreements

It is a common practice in the financial domain for a transaction between two or more parties to be governed by a single framework trading agreement, which provides the basic credit and relationship-level terms applicable to all such transactions between the parties. The role of such a framework agreement is commonly filled by master agreements, such as the ISDA Master Agreement in the case of over-the-counter derivatives. The ISDA Master Agreement has been referred to by Mr Justice Briggs in Lomas v JFB Firth Rixson Inc (2010) EWHC 3372 (Ch) as, "*probably the most important standard market agreement used in the financial world*".

The ISDA Master Agreement comprises a "preprint",[2] which is a standard part of the contract (other than the detailing of the parties to the contract, the "Dated as of date" and the signature block to the contract), and an accompanying Schedule. The Schedule (which has a published proforma form, although many financial institutions have their own "house-style") details elective terms (which require contracting parties to select between various predefined options), as well as any other bespoke modifications the parties to the preprint seek to make.

The commercial position achieved through the legal contract is therefore a combination of the terms of the preprint and the Schedule. Negotiators of the ISDA Master Agreement are very familiar with the relevant preprint forms, and therefore the split between the preprint and the Schedule allows faster negotiation of the commercial position between parties. It also assists with the overall standardization of language (through the fact that the preprint itself is effectively untouched in respect of the many aspects of the matters it covers). The boilerplate terms of the preprint provide benefits to market participants, reducing the number of "deal" points on which negotiating parties must contract, thereby reducing contracting costs and speeding up transactions (Choi and Gulati 2005).

Moving away from just the process of putting an agreement in place, the management of the contract lifecycle of an ISDA Master Agreement can be time-consuming and challenging, not only because the trading parties may have different business objectives and priorities, but because they may use different customized templates and negotiation guides to the Schedule, and use different systems to manage downstream business processes (such as risk management, capital and liquidity). It is fair to say that it has only been in the last fifteen years that market participants have truly recognized the importance of legal agreement data in respect of these contracts in order to manage the contractual obligations they contain. This has been mainly down to the 2007–2008 financial crisis, resulting in a need to better understand some of the specific (mainly credit-related) terms of executed ISDA Master Agreements, such as rating-downgrade events and cross-default clauses.

To address these challenges, ISDA, working with legal change and data consulting firm D2 Legal Technology (D2LT), launched the Clause Taxonomy and Library Project in 2018, "to identify provisions within the Schedule to the ISDA Master Agreement that may benefit from further standardization" (ISDA 2020). ISDA views increased standardization as a key component for developing enhanced legal documentation standards, and for facilitating further automation of derivatives products through the development of smart derivatives contracts. D2LT notes that the standardization of wording will better facilitate the path to automation in the OTC derivatives industry, by helping to define standard business processes in respect of contractual obligations (that are now more standardized in their drafting) contained

[2] It should be noted that at the time of writing, there are two forms of ISDA Master Agreement preprint published in 1987, two published in 1992—and a further 2002 form (ignoring the more recent Irish and French law versions). Of these five preprint forms, two of these forms (namely the "1992 ISDA Master Agreement (Multicurrency-Cross Border)" and "ISDA 2002 Master Agreement" account for more than 99% of the active ISDA Master Agreements in place (D2LT Legal Agreement Management Survey, conducted with 26 leading financial institutions in 2019).

within ISDA Master Agreement Schedules. In particular, the clause taxonomy creates an enumeration of the common business outcomes for each of the clauses in the ISDA Master Agreement when considering the agreement as a whole (i.e. ISDA Master Agreement preprint and negotiated Schedule).

This paper reports a parallel effort to D2LT's work on the Clause Taxonomy and Library Project (Datoo 2019, p. 79). In this report we analyse a representative sample of ISDA Schedule clauses specifying whether and under which conditions the AET provision of the ISDA Master Agreement applies. The goal of the analysis is to identify the underlying logic of the clauses, and to paraphrase the clauses in a form of Logical English. At the present time, this is a manual process, but one that can benefit from computer assistance in the future. By way of context, the operation of AET can be critical in respect of some ISDA Master Agreements, in terms of ensuring the operation of its close-out netting provisions (as detailed in Section 6 (Early Termination) of the ISDA Master Agreement).

3 Basic Logical English as syntactic sugar for logic programs

LE is a controlled natural language (Kuhn 2014), similar in spirit to Attempto Controlled English (ACE) (Fuchs and Schwitter 1996; Fuchs et al. 2008; Fuchs 2013), PENG (Schwitter, 2002) and PENG^{ASP} (Guy and Schwitter 2017). ACE and PENG are both intended for general-purpose knowledge representation and reasoning, and in that respect are more ambitious than LE. By comparison, LE and PENG^{ASP} are intended as syntactic sugar for logic programs. PENG^{ASP} provides syntactic sugar for the logic programming language ASP, and LE is syntactic sugar for the language LPS (Kowalski and Sadri 2015, 2016), which is an extension of logic programming (LP), implemented in Prolog (Wielemaker et al. 2019).

LPS extends LP by means of reactive rules, and the version of LE implemented by Davila (2017) includes an English syntax for such reactive rules as the following rule, where Bob transfers into Mary's bank account 10% of every amount that is transferred into his account:

> **If** an amount A is transferred into Bob's account from another account at a time T1
> **and** B = 0.10 × A
> **and** T2 is immediately after T1
> **then** B is transferred into Mary's account from Bob's account at T2.

Such reactive rules, which have the general form ***if*** *antecedent* ***then*** *consequent*, and which generate actions to make the *consequent* true whenever the *antecedent* becomes true, are not needed for the ISDA application. So, in the remainder of this paper we will restrict our use of the terms "Logical English" and "LE" to the LP component of LPS without reactive rules.

Although LE is intended for use as a general-purpose computer language, it is similar to such domain-specific English-like languages as Blawx (Morris 2020, 2021), Lexon (Diedrich 2020) and Oracle Intelligent Advisor (Lee 2020). Blawx and Lexon are both aimed at applying declarative computer language technology to legal applications, and Oracle Intelligent Advisor is aimed more generally at encoding rules for decision automation. LE, on the other hand, is inspired in part by the thesis that well-written legal documents "can be viewed as programs expressed in human language to be executed by humans rather than by computers" (Kowalski 1992).

The most basic form of LE is simply a sugared syntax for logic programs, where instead of writing symbolic expressions such as:

∀ X, Y, A, B1, B2, T1, T2 (account_balance(X, B2, T2) ←
(type(X, account), type(B2, amount), type(T2, time),
transfer(Y, X, A, T1), type(Y, account), different(X, Y),
type(A, amount), type(T1, time),
balance(X, B1, T1), type(B1, amount), sum(B1, A, B2), next(T1, T2)))

we write "controlled" English expressions such as:

The balance in an account is an amount B2 at a time T2
if an amount A is transferred into the account from another account at a time T1
and the balance in the account is an amount B1 at T1
and B2 = B1 + A
and T2 is immediately after T1.

In essence, the basic form of LE retains the top-level syntax of logic programs as consisting of atomic sentences and conditional sentences, and simply replaces the symbolic representation of an "atomic formula", such as transfer(Y, X, A, T1), having "parameters" (or "arguments") Y, X, A and T1, by an instance:

an amount A is transferred into the account from another account at a time T1

of a more general "template" containing variables (or "placeholders"):

an amount is transferred into **an account** from **another account** at **a time**.

Here variables (highlighted in bold font) are introduced by an indefinite article "a" or "an" or by the determiner "another" before a common noun, such as "account", which represents the type of the variable. In the symbolic representation, this type information has to be added separately.

The position of a variable in a template indicates its role in the template. Except for subjects and objects, these roles are indicated by the preposition preceding the variable (in this example, by one of the prepositions "into", "from" or "at"). In the symbolic representation, this role information is missing completely. For example, the symbolic expression parent(X, Y) gives no indication of the types of X and Y, nor does it tell which of X and Y is the parent and which is the child. In contrast, the template:

a person is a parent of **another person**

using bold font to highlight variables, not only provides all of this information, but it also indicates that the two persons are different and that the phrase "a parent" is not a variable.

In the same way that each instance of an atomic formula in formal logic needs to be expressed in the same form within a given formal theory, it is possible to insist that each instance of a template is expressed in LE in the same form within the same document. However, this restriction can be relaxed by allowing the roles of parameters signalled by prepositions to be expressed in any order (or even to be omitted when they are not relevant). For example, the two templates:

an amount is transferred into **an account** from **another account** at **a time**
an amount is transferred from **an account** into **another account** at **a time**

are interchangeable.

It is useful to name templates and collect them in a dictionary or "lexicon".[3] For example:

transfer: **an amount** is transferred into **an account** from **another account** at **a time**.

Having names for templates makes it possible, not only to assert sentences, but also to talk about sentences using nominalisation. This important feature of LE does not play any role in this paper, and we will not discuss this feature further in this paper.

Templates provide a standard representation for atomic sentences. It is also possible to standardise the syntax of verbs and nouns, for example by expressing all nouns and verbs in the singular. Restricting nouns to the singular means that LE does not use the English quantifier "all", which requires the use of a plural noun.

It is tempting to insist that all verbs are expressed in the active voice. But this is not always convenient, as in the case of a bank transfer performed by an intermediary whose identity might not be known or might not be relevant.

More importantly, by referring to time explicitly, it seems that verbs can be restricted to the present tense. For example, instead of saying that a fact held or that a fact will hold, it is possible to say that the fact holds at a time before now or that the fact holds at a time after now. Moreover, there is also a case to be made for avoiding the term "now" altogether (replacing it by a variable, whose value changes over time). But these are matters that require further investigation. So will not be insisted upon here.

[3] The dictionary also needs to include type declarations, for example to indicate that both Bob's account and Alice's account are types of accounts. We ignore this issue in this paper, because it is not important for the ISDA Master Agreement examples we investigate in this paper.

Sentences in LE are constructed from instances of templates. The simplest atomic sentences are "facts" that instantiate all parameters of a template by constants, as in:

£10 is transferred into Bob's account from Alice's account at 9:00/23/11/2020.

Atomic sentences can also contain variables, as in:

An amount is transferred into Bob's account from Alice's account at 9:00/23/11/2020.

Notice that it is natural to interpret such variables as "existentially quantified", as in:

Some amount is transferred into Bob's account from Alice's account at 9:00/23/11/2020.

Such existentially quantified variables have a wide scope that extends beyond the sentence in which they are introduced. For example:

The amount is greater than or equal to £10.

Rules in LE have the form of conditional sentences:

conclusion ***if*** *conditions*

where the *conclusion* is an atomic sentence, the *conditions* are a conjunction of atomic sentences or their negations; and an atomic sentence is an instance of a template.

In general, a parameter can be instantiated by a constant or by a variable.[4] The first use in a sentence of a variable of a given type is introduced by one of the indefinite articles "a", "an" or by the determiner "another" before a common noun, which represents the type of the variable. The variable can optionally be given a symbolic name, such as X, Y, B1, B2, A, T1, T2. If the variable has been given a name, then later references to the same variable in the same sentence are made simply by using the symbolic name of the variable. Otherwise, later references are made by replacing the indefinite article "a" or "an" by the definite article "the" before the same common noun, or by replacing "another" by "the other".

All variables in logic programs are implicitly "universally quantified", in the sense that variables stand for all their instances. Here is a representation of the sentence above that is closer to traditional logic, but still written in a form of controlled English:

For all Acct1, Acct2, B1, B2, T1, T2,
the balance in an account Acct2 is an amount B2 at a time T2
if an amount A is transferred into Acct2 from an account Acct1 at a time T1
and the balance in Acct2 is an amount B1 at T1
and B2 = B1 + A
and T2 is immediately after T1.

[4] Parameters in logic programs can also be instantiated by composite terms. The representation of such composite terms in LE needs further investigation.

All universally quantified variables, no matter how they are specified, are local to the sentence in which they occur, and can be reused in other sentences, without there being any relationship between the same variables occurring in different sentences.

However, variables in the conclusion of a rule that are not in the conditions of the rule are naturally interpreted as "existentially quantified", just like variables in atomic sentences that are not contained in rules.[5] For example:

An amount is transferred into Bob's account from Alice's account at a time
if the time is the beginning of the first day of a month.

The plan is to develop LE as a series of extensions, starting from the basic form, which closely mirrors the syntax of logic programs. For example, in an extended form of LE we can write:

The balance in an account **becomes** A + B
when an amount A is transferred into the account from another account
and the balance in the account is an amount B.

One of the guiding principles for these extensions is that they should not introduce ambiguities. In particular, pronouns, such as "he", "she", or "it", are not allowed; and in a given context every word should have only one meaning. For example, the following sentence is not allowed, even though, given appropriate background knowledge, the sentence is not ambiguous:

The balance in an account **becomes** A + B
when an amount A is transferred into **it** from another account
and the balance in **it** is an amount B.

Having sketched the main features of LE, it is important to point out the main difference between conditional sentences in LE and conditionals in imperative programming languages and in most business rules and expert systems languages. Conditionals in imperative programming languages typically have the form *IF conditions THEN do actions ELSE do other actions*; and in business rules and expert systems languages they have the form *IF conditions THEN do actions.* For example, *IF the balance in an account is an amount, and the amount is less than 0 THEN stop the account*. The difference lies in the semantics of declarative sentences in LE, which have truth values, and imperative sentences in other languages, which do not. As (Vranas 2008) puts it, "imperatives cannot be true or false, so they are shunned by logicians".

[5] Because all variables in logic programs are universally quantified, any existentially quantified variable in the conclusion of a rule is replaced by using a "skolem function", to name the variable as a function of any universally quantified variables in the same sentence. LE sweeps this technicality under the carpet, by leaving it to the underlying translation of LE into LP. Notice in this example that the amount transferred from Alice to Bob is a function of the month in which the transfer takes place. So different amounts can be transferred in different months.

The foundation of LE on LP was inspired in part by the similarity between LP rules and legal rules, which was first demonstrated in the 1980s by the representation of a major portion of the British Nationality Act as a logic program in Prolog (Sergot et al. 1986). The demonstration has been described as having been "hugely influential for the development of computational representations of legislation, showing how logic programming enables intuitively appealing representations that can be directly deployed to generate automatic inferences" (Prakken and Sartor 2015). This work led to further legal applications of Prolog, such as such as (Bench-Capon et al. 1987; Sergot et al. 1991). It has been influential in spurring the use of the Event Calculus (Kowalski and Sergot 1986) to capture the temporal dynamics of legal norms and legal facts (Marin and Sartor 1999), and the use of metalogics for modelling legal reasoning (Barlund and Hamfeld 1994). It also contributed to the development of an LP approach to rules and exceptions (Kowalski and Sadri 1991).

4 Rules and exceptions

In both ordinary natural language and legal language, it is common to express rules and exceptions in the form:

rule: conclusion **if** main conditions.
exception: **it is not the case that** conclusion **unless** additional conditions.

We will call this the *common form* for rules and exceptions. We will see that some of the Schedule clauses have this form. It is possible to represent such rules and exceptions in an extended form of LE, based on an extended form of LP (Satoh et al. 2010) that mirrors the common form. But it is also possible to represent them in the simpler form:

LE: conclusion **if** main conditions **and** additional conditions.

We will call this the *basic form*. It has the advantage that it can be implemented efficiently, because it is the standard form for rules in logic programs. Arguably, the basic form is often easier for a human to understand than the common form, because it can sometimes be hard to tell whether a sentence in the common form expresses a general rule or whether the sentence expresses an exception to the rule. Moreover, the rule and the exception might be separated by other clauses, making it harder to relate one sentence to the other.

On the other hand, the common form has the advantage that it separates the conditions under which a conclusion holds into the most important main conditions (expressed in the rule) and the less important additional conditions (expressed in exceptions).

There are other common forms for rules and exceptions, including the form:

rule: conclusion **if** common conditions.
exception: **however**, other conclusion **if** uncommon conditions.

We will see that Section 6a of the 1992 and 2002 ISDA Master Agreement preprints has this form, as do many of the Schedule clauses. This common form of rules and exceptions can also be represented either in an extended form of LE or in the basic form. Depending on the context, there can be several alternative ways to represent such rules and exceptions in the basic form. Here are two such alternatives:

LE1: conclusion **if** common conditions **and it is not the case that** uncommon conditions.
other conclusion **if** uncommon conditions.

LE2: conclusion **if** common conditions **and it is not the case that** other conclusion.
other conclusion **if** uncommon conditions.

The common and basic forms for rules and exceptions are equivalent in the underlying "non-monotonic" logic of LP, which employs a "closed world assumption" that:

it is not the case that a sentence holds
if it cannot be shown that the sentence holds.

The closed world assumption is similar to the informal logic we often apply in practice when we interpret information in a relational database, assuming that "the only possible instances of a relation are those implied by the database" (Reiter 1989). Moreover, it formalises the intuitive way that we commonly understand many natural language conditionals (Stenning and Van Lambalgen 2012).

5 Automatic early termination

In the remainder of this report, we investigate the Automatic Early Termination provisions of the 1992 and 2002 ISDA Master Agreement preprint, followed by a sample of Schedule clauses specifying whether and under which conditions Automatic Early Termination applies. The goal of this investigation is to explore the extent to which the use of LE can contribute to the standardisation and potentially to the automation of the contractual provisions of important financial contracts such as the ISDA preprint and its Schedule.

The real agreement samples investigated in this report were provided by D2LT as representative of the range of AET Schedule clauses seen in an exercise reviewing many thousands of ISDA Master Agreements in place between various market participants, in order to make close-out netting enforceability determinations (Datoo, 2019, page 102). The AET clause can often have a material impact on such determinations, which has a significant impact in turn on the regulatory capital calculations

of prudentially regulated financial institutions. In fact, regulatory capital optimisation is one of the main reasons for the amendment of the AET clause.

Taken in their financial context, the Event of Default provisions[6] of the ISDA Master Agreement are probably the most important in the entire agreement, as they enable a party to terminate (or "close-out") all outstanding Transactions governed by the Agreement in certain circumstances, typically when there is a significant risk that the other party will not perform under the Agreement. Section 6(a) of the ISDA Master Agreement preprint provides that when an Event of Default occurs, the Non-Defaulting Party may, at any time when the event is still continuing, close-out the Agreement and all outstanding Transactions under the Agreement. This close-out is given effect by the Non-Defaulting Party giving notice to the Defaulting Party, designating the date on which such close-out will occur (an "Early Termination Date"). This notice does not need to be delivered immediately, but the Early Termination Date must be up to, but no more than, twenty days after the service of the notice.

One of the elections made in the Schedule of an ISDA Master Agreement regards the applicability of Automatic Early Termination. If this election is made with respect to a party to the ISDA Master Agreement, then the occurrence of certain Bankruptcy Event of Default events (a subset of those specified in Section 5(a)(vii) of the Bankruptcy Event of Default) with respect to the party (the "Defaulting Party" in this case) will automatically cause all outstanding Transactions under the ISDA Master Agreement to be closed-out (rather than requiring the Non-Defaulting Party to send a notice specifying the Early Termination Date for the close-out to occur).

The purpose of applying Automatic Early Termination is to ensure that close-out takes place before the commencement of bankruptcy proceedings, as the insolvency laws of certain jurisdictions may mean that any attempt to close-out after such proceedings is ineffective, despite the contract stating otherwise. This is currently the case in jurisdictions such as South Africa and Switzerland. Automatic Early Termination seeks to overcome such issues by providing that the close-out occurs whilst it might still be effective.

These issues highlight the advantages of applying the Automatic Early Termination provisions in jurisdictions where the insolvency laws need such consideration. However, automatic termination does have commercial disadvantages. In particular, the Non-Defaulting Party may not necessarily become aware that the Agreement and the Transactions thereunder have been closed out until some time after the close out has occurred. This may expose the parties to significant risk of market movements for which they are unhedged.

[6] Many of the capitalised terms not defined in this discussion, such as "Event of Default", are from, and defined within, the text of the ISDA Master Agreement preprint itself.

6 Early termination in Section 6(a) of the master agreement

Before analysing the logic of the Schedule clauses, we need to analyse the logic of Section 6(a) of the preprint forms itself. To ease the burden of understanding, the formatting has been removed from the preprint, spacing has been added and the most important logically significant keywords have been highlighted in **bold font**. Notice that Section 6(a) is expressed in one of the common forms for a rule and exception:

> 6(a) Right to Terminate Following Event of Default. **If** at any time an Event of Default with respect to a party (the "Defaulting Party") has occurred **and** is then continuing, the other party (the "Non-defaulting Party") may, by not more than 20 days notice to the Defaulting Party specifying the relevant Event of Default, designate a day not earlier than the day such notice is effective as an Early Termination Date in respect of all outstanding Transactions.
> **If, however**, "Automatic Early Termination" is specified in the Schedule as applying to a party, **then** an Early Termination Date in respect of all outstanding Transactions will occur immediately upon the occurrence with respect to such party of an Event of Default specified in Section 5(a) (vii) (1), (3), (5), (6) or, to the extent analogous thereto, (8), **and** as of the time immediately preceding the institution of the relevant proceeding or the presentation of the relevant petition upon the occurrence with respect to such party of an Event of Default specified in Section 5(a)(vii)(4) or, to the extent analogous thereto, (8).

The LE representation of the logic of the first sentence of 6(a) illustrates many of the most important features of the basic form of LE:

LE: **It is permitted that** a party designates by a notice at a time T2 to another party that an Early Termination Date in respect of a Transaction[7] occurs at a time T3
if the Transaction is outstanding at T3
and an Event of Default occurs with respect to the other party at a time T1
and the Event of Default is continuing at T2
and the notice specifies the Event of Default
and $T2 \leq T3$ and $T3–T2 \leq 20$ days
and it is not the case that the Schedule specifies that Automatic Early Termination applies to the other party for the Event of Default.

Here the logical keywords highlighted in bold font demarcate the separate template instances, whose templates with their parameters would be specified in a separate dictionary.

[7] This formulation, with "a Transaction" in the conclusion and with the condition "the Transaction is outstanding at T3" is logically equivalent to having instead "each outstanding Transaction" in the conclusion, which is closer to the original English "all outstanding Transactions".

Notice that, in the conclusion of the LE sentence, the parties referred to by the phrases "a party" and "another party" are opposite to the parties referred to in the original English sentences. However, the two sentences have the same meaning.

The last condition of the rule prevents the rule from being applied if the exceptional condition specified in the second sentence of 6(a) applies. For the purposes of standardisation, the LE template:

> the Schedule specifies that Automatic Early Termination applies to **a party**
> for **an Event of Default**

for the exceptional condition uses the present tense of the active voice of the verb "specifies" instead of the original verb "is specified" in the passive voice. In comparison with the original English, it also includes an extra parameter "**an Event of Default"**. We will see later, when we investigate the representation of the Schedule clauses, why this extra parameter is added. In the meanwhile, it is worth noting that AET is possible only for a subset of all Events of Default, namely only for certain bankruptcy Events of Default.

The condition before last is expressed as a mathematical inequality. It could also be expressed more verbosely, for example as:

> T3 is at the same time or after T2
> **and** T3 is at the same time or before T2 + 20 days.

The phrase "in respect of" in the conclusion and the phrase "with respect to" in the first condition function as prepositions that identify the role of the following parameter.

The second and third conditions of the sentence retain the distinction of the original English sentence between the condition that an Event of Default has occurred and the condition that the Event is continuing. However, in LE, events do not literally continue. Instead, LE employs the ontology of the Event Calculus (EC) (Kowalski and Sergot 1986), in which events occur at time points, and events initiate or terminate "fluents", which are facts that flow with time. In accordance with the EC ontology of LE, the third condition of the sentence that the Event of Default is continuing at time T2 concerns a fluent, which represents both the ongoing **state** of Default and the identity of the earlier Event of Default that initiated the state of Default. The fluent can be terminated, for example, by "curing" the default, in which case the Event of Default would no longer be "continuing".

The second sentence of 6(a) specifies the occurrence and timing of an Early Termination Date, which takes place automatically if AET is specified in the Schedule (as applying to a party) upon the occurrence of certain bankruptcy Events of Default. In contrast, the first sentence merely permits the Non-defaulting Party to designate the occurrence of such an Early Termination Date upon the occurrence of any kind of Event of Default.

Permission, along with obligation and prohibition, is a deontic notion, which is normally formalised by means of a specialised deontic logic. However, the metalogical (or "higher-order'') capabilities of LP, which are inherited by LE, are sufficiently powerful to represent such deontic relationships, e.g. (Sergot 1982;

Lokhorst 1996). In the case of Section 6a and similar cases, the deontic relationship between a party being permitted to designate the occurrence of an event, and the actual occurrence of the event can be expressed in LE as[8]:

An event occurs at a time T2
if it is permitted that a party designates by an action at a time T1
that the event occurs at T2
and the party performs the action at T1.

The LE representation of the second sentence consists of two rules for two cases:

LE: An Early Termination Date occurs in respect of a Transaction at a time T
if the Transaction is outstanding at T
and an Event of Default of type Section 5(a) (vii)(1), (3), (5), (6) or, to the extent analogous thereto, (8) occurs for a party at T
and the Schedule specifies that Automatic Early Termination applies to the party for the Event of Default.

An Early Termination Date occurs in respect of a Transaction at a time T
if the Transaction is outstanding at T
and an Event of Default of type Section 5(a) (vii) (4) or, to the extent analogous thereto, (8) occurs for a party at a time T1
and the institution of the relevant proceeding or the presentation of the relevant petition for the Event of Default occurs at a time T2
and the Schedule specifies that Automatic Early Termination applies to the party for the Event of Default
and T is immediately before T2.

The first rule deals with the case of a Bankruptcy Event of Default specified in Section 5(a) (vii) (1), (3), (5), (6) or, to the extent analogous thereto, (8); and the second one with the case of a Bankruptcy Event of Default specified in Section 5(a) (vii) (4)[9] or, to the extent analogous thereto, (8). The conditions for the first case can be satisfied in five different ways; and the conditions for the second case in two different ways for the second condition and two different ways for the third condition. The

[8] This sentence has a simple representation as a Prolog clause:
happens(Event, T2):- permitted(Agent, designate(Action, T1, Event, T2)), happens(Action, T1).

[9] This second kind of bankruptcy is one where a party institutes or has instituted against it, a bankruptcy proceeding or petition.

unravelling of the two rules into nine different rules for the nine different subcases is a minor complication, which is easily solved using standard LP meta-programming techniques, which we can ignore in this paper.

The treatment of "vague" conditions, such as the condition that one type of event is "analogous" to another type of event, is also fairly well-understood. There are several different ways in which such vague conditions can be evaluated: (1) by human judgement that the condition holds, (2) by additional rules specified in the legal document itself, (3) by rules formulated by an expert advisor, or (4) by rules generated by machine learning from a training set of cases.

But no matter how such vague conditions are evaluated at a lower level, this way of decomposing problems into subproblems, from higher levels to lower levels, top-down, is a standard LP methodology. It has the advantage that you can defer dealing with subproblems until after you have addressed the more important problems at the higher level. In this paper, we have taken advantage of the top-down approach to defer until another occasion the logical analysis of some of the complex conditions of the rules that we investigate in this paper.

We have also deferred consideration of some of the conclusions of the rules, as in the case of the conclusion "An Early Termination Date occurs in respect of a Transaction at a time T" of the two rules above. The consequences of early termination are specified by other rules in the preprint.

Another issue, and one that is less familiar in LP circles, is the logical status of a variable (signalled by "a" or "an") in the conclusion of a rule that is not also in the conditions of the rule (where it would be signalled by "the"). This is the case with the variable "an Early Termination Date" in the conclusion of the two LE rules above.

In conventional LP languages, all variables, whether in the conclusion or in the conditions of a rule, are implicitly universally quantified, standing for all of their instances. Understood in this conventional LP manner, the conclusion of the two LE rules above would mean that:

each Early Termination Date occurs in respect of a Transaction at a time T.

This is not the natural interpretation of the variable "an Early Termination Date" in these two rules. As we have already argued in the case of similar examples, it is more natural to interpret the variable existentially, as we do in LE.

The last condition of the last LE rule (T is immediately before T2) is possibly confusing, since it literally means that an Early Termination Date occurs at **each** time T that is immediately before T2. Of course, in this context there would be only one such time T, and its determination would be a matter of legal interpretation. Moreover, the time T is a function of T2. In a functional programming language, mixed functional-relational language or an extended version of LE, the condition would be inserted into the conclusion. The resulting representation would be closer to natural language, and would be easier to read. Here is how the sentence might look in such an extended LE, where a restrictive relative clause is used to insert the condition into the conclusion of the sentence, and "the" is used to signal that there is only one variable of the relevant type:

An Early Termination Date occurs in respect of a Transaction
at **the time T that** is immediately before a time T2
if the Transaction is outstanding at T
and an Event of Default of type Section 5(a) (vii) (4) or, to the extent analogous thereto, (8) occurs for a party at a time T1
and the institution of the relevant proceeding or the presentation of the relevant petition for the Event of Default occurs at T2
and the Schedule specifies that Automatic Early Termination applies to the party for the Event of Default.

Notice that the actual time T1 of the Event of Default is not relevant. It is the time T2 of the institution of the relevant proceeding or the presentation of the relevant petition that determines the time T of the occurrence of AET. In an extended LE, the time parameter T1 could be omitted and the second condition could be written more simply as:

an Event of Default of type Section 5(a)(vii)(4) or, to the extent analogous thereto, (8) occurs for a party.

7 Automatic early termination in the schedules

In this section, we consider a sample of clauses specifying the application (or non-application) of AET in a variety of ISDA Schedules. The clauses are presented in order of increasing complexity. To ease understanding, spacing has been added to the original English text, and the most important logical keywords have been highlighted in **bold**. Moreover, logical keywords that are only implicit in the original English text have been made explicit, by adding and highlighting them in ***bold italic font***.

By way of background, it is useful to note that the ISDA Master Agreement is drafted in a manner that expects the two trading parties to the agreement to be assigned a label as either "Party A" or "Party B". Despite the fact that some financial institutions prefer to assume the title of Party A (rather than the title of Party B), there is no actual advantage or disadvantage from doing so. It is purely definitional (as perhaps one might argue many titles are!).

The fact that the two parties to the contract are defined as Party A and Party B can be represented in LE by the atomic sentences:

Party A is a party.
Party B is a party.[10]

[10] "a" in the context "is a" does not signal the occurrence of a variable, but rather identifies the type of the individual referred to before the "is a".

These LE clauses are type declarations. In the context of the closed world assumption, they imply that there are only two trading parties, Party A and Party B.

Example Clause 1.

English: The "Automatic Early Termination" provision of Section 6(a) will apply to Party A **and** will apply to Party B.

This is one of the simplest cases. But, to be computer processable, the Clause needs to match the corresponding template in the LE representation of 6(a), which has the form:

the Schedule specifies that Automatic Early Termination applies to **a party** for **an Event of Default**.

The need for the additional parameter "an Event of Default" will be seen when we consider example Clause 8. In the meanwhile, we note that the need for the parameter is analogous to the need to be more precise if we want to say, for example, that Jumbo, an elephant, is big, because Jumbo can be big for a typical animal, but small for an elephant. In the case of AET, we similarly need to be precise about whether it applies only to the first case of Events of Default of type Section 5(a) (vii) (1), (3), (5), (6) or, to the extent analogous thereto, (8), only to the second case of Events of Default of type Section 5a (vii) (4) or, to the extent analogous thereto, (8), or to both cases.

In situations of the kind in Clause 1, where AET applies to both cases, we can represent the application of AET in a simplified form (analogous to saying simply that Jumbo is big without qualification) by employing a simplified template:

the Schedule specifies that Automatic Early Termination applies to **a party.**

The gap between the simplified template of the Schedule and the more precise template of the preprint can be bridged by means of a linking rule in LE:

LE: The Schedule specifies that Automatic Early Termination applies to a party for an Event of Default

if Automatic Early Termination applies to the party

and the Event of Default of type Section 5(a) (vii) (1), (3), (4), (5), (6) or, to the extent analogous thereto, (8) occurs for the party at a time.

The last condition of the linking rule ensures that the variable "an Event of Default" in the conclusion of the rule is suitably qualified, so that it is neither existentially quantified (meaning that it applies to **some** Event of Default) nor universally quantified without qualification (meaning that it applies to **every** event of default without restriction). Notice that the simplified LE language of the Schedule uses the present tense "applies" in place of the original English "will apply".

The variable "a/the party" in the linking rule can be instantiated to only one instance at a time. So to represent Clause 1, we need two separate LE sentences, in addition to the linking rule:

LE: Automatic Early Termination applies to Party A.
Automatic Early Termination applies to Party B.

A more compact way (which we will employ in some of the later examples) is to represent Clause 1 by a more general LE sentence:

LE: Automatic Early Termination applies to each party.

meaning that AET applies to all parties, but avoiding the use of the plural noun "parties". If "each party" were replaced by "a party" it would be interpreted by the LE to LP translator existentially as meaning that there is some party to which AET applies, which is not faithful to the meaning of the original English sentence.

Example Clause 2.

English: The "Automatic Early Termination" provision of Section 6(a) will **not** apply to Party A **and** will **not** apply to Party B.

In theory, this non-applicability clause is entirely unnecessary. It is logically adequate to say nothing at all in the Schedule about the application of AET to Parties A and B. This is because both the original English sentence 6(a) and the LE representation of 6(a) require only that the Schedule specifies that the AET provision applies if the AET provision does apply. The closed world assumption implies that the AET provision does not apply if the Schedule does not specify that it does apply.

However, it would be necessary to specify in the Schedule that AET does not apply if the relevant implicit condition in the first English sentence of 6(a) were:

> the Schedule specifies that Automatic Early Termination **does not apply** to the other party for the Event of Default

instead of

> **it is not the case that**
> the Schedule specifies that Automatic Early Termination **applies** to the other party for the Event of Default.

It's like the difference between believing there is no Santa Claus and not believing there is a Santa Claus. Or between "this paper assumes no previous background in formal logic, computing, finance or law" and "this paper does not assume any previous background in formal logic, computing, finance or law". It's easy to be confused.

Although the inclusion of a non-applicability clause in the Schedule is logically unnecessary, it is harmless, unless one considers the extra, unnecessary computational effort needed to process such unnecessary sentences. Nonetheless, its

inclusion shows that the commercial implications of applying AET have been considered, and as a result a decision has been taken not to apply AET.

Example Clause 3.

English: The "Automatic Early Termination" provision of Section 6(a) of this Agreement will **not** apply to Party A **and** will apply to a Party B **if** identified as applicable on Appendix I (as periodically amended).

Before we look at an LE representation of this clause, we provide some background to this clause, which is taken from a particular type of ISDA Master Agreement often referred to as an umbrella ISDA Master Agreement. This is an ISDA Master Agreement that deems a number of identical ISDA Master Agreements to be created between (typically) one Party A and a number of Party Bs, with a list of such Party Bs provided in an Appendix.

In the case of this Example Clause 3, as well as Appendix I providing a list of such Party Bs, the Appendix also details for each Party B whether AET will apply to Party B (which is likely to be determined by the jurisdiction of incorporation of Party B, and the insolvency laws of the jurisdiction, and therefore whether AET is required for the likely enforceability of the close-out provisions of an ISDA Master Agreement in the event of the bankruptcy (or events pertaining to bankruptcy) of Party B).

As mentioned above, the first part of the clause, stating that AET does not apply to Party A, is logically unnecessary, but harmless. The second part of the clause, stating a condition under which AET applies to Party B, is already close to an LE formulation. It is necessary only to add the missing implicit parts of the condition of the clause:

LE: The Schedule specifies that Automatic Early Termination applies to Party B for an Event of Default
if Appendix I (as periodically amended) specifies that
Automatic Early Termination applies to Party B for the Event of Default.

Notice that in this formulation, the conclusion of the rule directly matches the template used in the LE representation of 6(a). So there is no need for the linking rule used for Clause 1.

Here the qualification "as periodically amended" can be omitted as superfluous, because if an amendment is effective, there is no need to say so. However, it does highlight that the Appendix is likely to be updated from time to time for commercial reasons.

Example Clause 4.

English: The "Automatic Early Termination" provision of Section 6(a) will **not** apply to Party A **and** will, **depending** on netting opinions, apply to Party B.'

Again, the first part, concerning Party A, is logically unnecessary, but harmless. The second part of the clause, concerning Party B, is already close to an LE formulation. For example:

LE: The Schedule specifies that Automatic Early Termination applies to Party B for an Event of Default
if netting opinions advise that Automatic Early Termination applies to Party B for the Event of Default.

Netting opinions analyse the insolvency laws in relevant jurisdictions and guide the parties whether to apply Automatic Early Termination to try to ensure that the close-out netting provisions will be viewed as enforceable by a court of law. This guidance is given to the parties to advise them during the negotiation of an agreement,[11] rather than as a condition to be inserted into the final agreement. So it is likely that this example was included in the real agreement samples by mistake.

Notice, by the way, that the plural noun and verb in the phrase "netting opinions advise" violate our intention to standardise on the use of singular nouns and verbs. Moreover, the use of the plural is more vague than the singular. The meaning of the condition expressed in the singular:

> a netting opinion advises that Automatic Early Termination applies to Party B for the Event of Default.

has a clear meaning (whether or not it is the meaning that is intended), whereas the same condition expressed in the plural does not have a clear meaning.

Example Clause 5.

English: The "Automatic Early Termination" provision of Section 6(a) will **only** apply to Party A and to Party B **if** the laws of a jurisdiction other than the laws of the United States applies to this Master Agreement, the Credit Support Annex, or the collateral under the Credit Support Annex.

This is a rare form of wording for an Automatic Early Termination clause in an ISDA Schedule. It appears that the parties very much intended the laws of the United States[12] to be the only relevant applicable laws, and it is likely that the wording reflects the identities and geographic locations of the two parties and their nexus only to the laws of the United States. They have, almost lazily in drafting, simply switched on "Automatic Early Termination" as a (commercially) defensive step in case it might be necessary to deal with insolvency laws outside of the United States.

In traditional treatments of formal logic, an English expression of the form *p only if q* implies that *p* cannot hold unless *q* holds, and if *p* holds it must therefore be

[11] There is ongoing research into the standardisation and automation of specific operational aspects of this close-out netting determination process by reference to legal opinions (Datoo and Clack, 2021).

[12] Also note the reference to the laws of the United States rather than relevant state laws, which we ignore for the purposes of this paper.

true that *q* holds. Thus in traditional treatments of formal logic we have the normal translation from *p only if q* into the logical expression *q if p* (or equivalently into *if p then q*). But the statement *q if p* taken alone gives only hypothetical information about what would be the case if *p* were to hold, not whether *p* actually holds. Thus, if this translation were applied to Clause 5, it would give only hypothetical information about what would be the case if AET were to apply. It would give absolutely no information about whether or not AET actually does apply.

In contrast, the underlying non-monotonic logic of LE translates the English expression *p only if q* in situations like this into *p if q*, together with a further closed world assumption that *p if q* is the ***only*** way of showing that *p* is the case, because there are no other rules with conclusion *p*. Applying this understanding of *p only if q*, gives an LE sentence of the form:

LE: Automatic Early Termination applies to each party
if the laws of a jurisdiction other than the laws of the United States apply to this Master Agreement, the Credit Support Annex, or the collateral under the Credit Support Annex.

Here the conclusion of the rule is stated using the simplified template that needs the linking rule used for Clause 1.

Example Clause 6.

English: The "Automatic Early Termination" provision of Section 6(a) will **not** apply to Party A and will **not** apply to Party B, **provided, however**, that **where** the Event of Default specified in Section 5(a) (vii) (1), (3), (4), (5) or (6), or to an analogous extent, (8), is governed by a system of law which does not permit close-out netting to take place after the occurrence of the relevant Event of Default, **then** the Automatic Early Termination provisions of Section 6(a) shall apply.

This Example Clause 6 generalises Clause 5, in the sense that Clause 5 excludes the jurisdiction of the United States, whereas Clause 6 has no exclusions on the jurisdiction. However, Clause 6 is also more specific than Clause 5, because it deals with the insolvency law analysis that might lead to the sensible application of Automatic Early Termination (rather than assuming, as in the case of Clause 5, that insolvency laws outside of the United States would require Automatic Early Termination to apply to the parties for close-out netting purposes).

However, the sentence is ambiguous. It is not clear whether the proviso applies to both Parties A and B or only to Party B. However, if it were intended to apply only to Party B, then it would have been better if there were a comma, or perhaps a semicolon, between the phrases "will not apply to Party A" and "will not apply to Party B". In any case, it would have been still better to avoid all mention of AET not applying and to focus instead on specifying when AET does apply. Assuming the

interpretation that the proviso applies to both parties, the clause can be represented as follows:

LE: The Schedule specifies that Automatic Early Termination applies to a party for an Event of Default
if the Event of Default of type Section 5(a) (vii) (1), (3), (4), (5), (6) or, to the extent analogous thereto, (8) occurs for the party at a time T1
and the Event of Default is governed by a system of law which does not permit close-out netting to take place at a time T2
and T2 is after T1.

Note that the conclusion of the rule uses the full template for the applicability of AET with the qualification "for an Event of Default". Alternatively, the conclusion could use the simplified template without the qualification, together with the rule linking the two templates. The two alternatives are logically equivalent in this example.

Example Clause 7.

English: The "Automatic Early Termination" provision of Section 6a will **not** apply to Party A **or** Party B, **provided, however**, that **if** at any time an Event of Default specified in Section 5(a) (vii) (1), (3), (4), (5), (6) or, to the extent analogous thereto, (8), with respect to a party has occurred **and** is then continuing, **and** any court, tribunal or regulatory authority with competent jurisdiction acting pursuant to any bankruptcy or insolvency law or other similar law affecting such party makes an order which has or purports to have the effect of prohibiting the other party from designating an Early Termination Date in respect of all outstanding Transactions at any time after such Event of Default has occurred and is then continuing, in accordance with Section 6(a), ***then*** the "Automatic Early Termination" provision of 6(a) will apply to such party.

The purpose of this clause is to trigger Automatic Early Termination if the law attempts to prevent discretionary termination. It is very similar in effect to Example Clause 6, other than it does not just require the relevant laws to provide for the prevention of such termination after certain bankruptcy events have occurred, they actually need to take effect. So, if an Event of Default to which AET might apply occurs at time T1 and is continuing at time T2, but the law prevents the non-defaulting party from specifying an Early Termination Date, then early termination is triggered either at T1 or immediately before T1, depending on which limb of AET applies. Notice that the time of the order preventing the discretion from being exercised could take place after the Event of Default at time T1. So, the clause can have a "back in time" retroactive effect.

Ignoring the unnecessary specification of non-applicability, and the internal structure of the clause's complex condition, the clause has a simple top-level structure:

LE: The Schedule specifies that Automatic Early Termination applies to a party for an Event of Default
if the Event of Default of type Section 5(a) (vii) (1), (3), (4), (5), (6) or, to the extent analogous thereto, (8) occurs for the party at a time T1
and the Event of Default is continuing at a time T2
and a court, tribunal or regulatory authority with competent jurisdiction acting pursuant to any bankruptcy or insolvency law or other similar law affecting the party makes an order which has or purports to have the effect of prohibiting another party from designating an Early Termination Date for the Event of Default in respect of all outstanding Transactions at T2.

In theory, the internal logical structure of the complex condition can be deconstructed and represented explicitly, using the metalogical, metaprogramming features of the logic underpinning LE. But in practice, this would probably be unnecessary, because the judgement that the condition holds would probably be made by human input and be recorded as a separate statement of fact.

Example Clause 8.

English: The "Automatic Early Termination" provision of Section 6a will apply to both parties **subject to adding** at the end thereof the following words: "**provided, however,** that with respect to an Event of Default specified in Section 5(a)(vii)(4) or, to the extent analogous thereto, (8), **the second sentence of this** Section 6(a) **shall only apply if** the relevant proceeding is instituted by, or the relevant petition is presented to, a court or authority in the jurisdiction where the Defaulting Party is incorporated."

Notice that this style of drafting is different from the others in that, instead of specifying the logical conditions under which AET applies, it seeks to amend the preprint wording, albeit in the Schedule, rather than in the preprint itself. This lack of consistency in different styles of drafting is a problem in its own right. It can make it harder for the two parties negotiating the Schedule to reach an agreement (even if they are aligned on the business outcome itself), and it makes it harder to automate the processing of such agreements.

To illustrate the combined, net effect of the preprint and this particular Schedule clause, the same business outcome would have been reached if the Schedule clause had simply stated that:

> The "Automatic Early Termination" provision of Section 6(a) will apply to both parties.

and the second sentence in Section 6a of the preprint were rewritten (bold added to show the change from its standard form presented earlier):

> 6(a) Right to Terminate Following Event of Default. …
> …
> If, however, "Automatic Early Termination" is specified in the Schedule as applying to a party, then an Early Termination Date in respect of all outstanding Transactions will occur immediately upon the occurrence with respect to such party of an Event of Default specified in Section 5(a) (vii) (1), (3), (5), (6) or, to the extent analogous thereto, (8), and as of the time immediately preceding the institution of the relevant proceeding or the presentation of the relevant petition upon the occurrence with respect to such party of an Event of Default specified in Section 5(a) (vii) (4) or, to the extent analogous thereto, (8) **provided, however, that with respect to an Event of Default specified in Section 5(a) (vii) (4) or, to the extent analogous thereto, (8), the second sentence of this Section 6a shall only apply if the relevant proceeding is instituted by, or the relevant petition is presented to, a court or authority in the jurisdiction where the Defaulting Party is incorporated.**[13]

From this rewriting, one will note that this Example Clause 8 simply further qualifies the subtypes of Bankruptcy Events of Default to which Automatic Early Termination applies. In the following LE representation, the desired amendment is obtained simply by adding an extra condition to the third LE clause representing 6(a), bearing in mind that here, similarly to the case of Clause 5, *p only...if q* means that *q* is a necessary condition for concluding *p*:

LE: Automatic Early Termination applies to each party.

Amendment of the third LE clause representing Section 6(a):

LE: An Early Termination Date occurs in respect of a Transaction at a time T
if the Transaction is outstanding at T
and an Event of Default of type Section 5(a) (vii) (4) or, to the extent analogous thereto, (8) occurs for a party at a time T1

[13] Notice that this form of wording is rather strange (although its intention is clear). It adds a reference to the second sentence without taking into account that the reference occurs in the second sentence itself. Moreover, what does it mean to say that "a sentence applies"?

and the institution of the relevant proceeding or the presentation of the relevant petition for the Event of Default occurs at a time T2

and the Schedule specifies that Automatic Early Termination applies to the party for the Event of Default

and T is immediately before T2

and the institution of the relevant proceeding by or the presentation of the relevant petition to a court or authority occurs in the jurisdiction where the party is incorporated.

This approach, amending the LE representation of the preprint, requires less indirection, but defeats the very purpose of a preprint style agreement and standard form. There is an alternative approach, which involves no amendments, and is closer to the approach used in the LE representations of the earlier Clauses 1–6. Instead of incorporating the extra, necessary condition into an amendment, the alternative approach incorporates the extra condition into the definition of the applicability of AET in the Schedule itself. For this purpose, we need to split the definition into two rules, corresponding to the two types of bankruptcy Events of Default referred to in the second sentence of 6(a):

LE: The Schedule specifies that Automatic Early Termination applies to a party for an Event of Default

if the Event of Default of type Section 5(a) (vii) (1), (3), (5), (6) or, to the extent analogous thereto, (8) occurs for the party at a time T.

The Schedule specifies that Automatic Early Termination applies to a party for an Event of Default

if the Event of Default of type Section 5(a) (vii) (4) or, to the extent analogous thereto, (8) occurs for the party at a time T1

and the institution of the relevant proceeding or the presentation of the relevant petition for the Event of Default occurs at a time T2

and the institution of the relevant proceeding by or the presentation of the relevant petition to a court or authority occurs in the jurisdiction where the party is incorporated.

The representation illustrates the use of the qualification "for an Event of Default" in the template representing the applicability of AET to link the conclusions of the two rules with the type of Event of Default in the conditions of the rules. To see the need for the qualification, consider the following simplified representation without the qualification:

LE: A Case 1 Early Termination Date occurs
if a Case 1 Event of Default occurs **and** the Schedule specifies AET.
A Case 2 Early Termination Date occurs
if a Case 2 Event of Default occurs **and** the Schedule specifies AET.
The Schedule specifies AET **if** a Case 1 Event of Default occurs.
The Schedule specifies AET **if** a Case 2 Event of Default occurs
and other conditions.

Suppose two Events of Default occur: One of them is a Case 1 Event of Default; and the other is a Case 2 Event of Default, but **without the other conditions** for applying AET to a Case 2 Event. Then the occurrence of the Case 1 Event of Default will imply the occurrence not only of a Case 1 Early Termination Date, but also the unintended occurrence of a Case 2 Early Termination Date. The addition of the extra parameter for an Event of Default to the template for the specification of AET prevents this unintended consequence.

Example Clause 9.

English: The "Automatic Early Termination" provisions of Section 6a will apply to Party A **and** will **not** apply to Party B; **provided, however**, that with respect to a party, **where** the Event of Default is specified in Section 5(a) (vii) (1), (3), (5), (6) or, to the extent analogous thereto, (8), is governed by a system of law which does not permit termination to take place after the occurrence of the relevant Event of Default, **then** the Automatic Early Termination provisions of Section 6(a) will apply to such party. **Notwithstanding the foregoing**, with respect to any Insured Transaction, the "Automatic Early Termination" provision of Section 6a shall **not** apply to Party B **unless** an Additional Termination Event[14] set forth in Part 1(j)(ix)(a) of this Schedule has occurred.

This is undoubtedly the most problematic of the example clauses. The first problem is the conflict between the initial statement that AET applies to Party A, but not to Party B, and the following general statement, which has the form "**provided, however**, that with respect to **a party**" **certain conditions hold for the party**. There are two logical interpretations (noting that, applying commercial reality to the situation, interpretation 2 is likely to prevail):

[14] The ISDA Master Agreement creates two types of events that might terminate transactions. The first of these is an Event of Default, which is serious enough to result in the termination of all Transactions between the parties, should the non-defaulting party decide to designate an Early Termination Date (or if it is arrived at due to the application of Automatic Early Termination). The second is a "Termination Event". This is typically regarded as less serious than an Event of Default (or, the relevant party is less culpable for its occurrence than would typically be the case for the Event of Default, e.g. an Illegality). Accordingly, the consequences of the two events are different. The occurrence of an Early Termination Date related to an Event of Default results in the termination of all Transactions, whereas in the case of a Termination Event, only certain "Affected Transactions" may be impacted.

1. AET applies to a party if certain conditions hold for the party.
2. AET applies to Party A.
   AET applies to Party B if certain conditions hold for Party B.

In the first interpretation, the initial statement that AET applies to Party A, but not to Party B, is unnecessary, unhelpful and misleading. In this case, the LE representation of Clause 9 consists of two clauses, one for Party A, and the other for Party B. The LE representation for Party B will be considered separately. But, so far as Clause 9 concerns Party A, the entire Clause can be represented by:

LE: The Schedule specifies that Automatic Early Termination applies to Party A for an Event of Default
if the Event of Default of type Section 5(a) (vii) (1), (3), (5), (6) or, to the extent analogous thereto, (8) occurs for Party A at a time T1
and the Event of Default is governed by a system of law
which does not permit termination to take place at a time T2
and T2 is after T1.

In the second interpretation, the phrase "**a party**" in the qualification is too general, and should be replaced by "**Party B**". Also the phrase "**then** the Automatic Early Termination provisions of Section 6a will apply to **such party**" should be replaced by "**then** the Automatic Early Termination provisions of Section 6(a) will apply to **Party B**". In this case, in relation to Party A, the entire Clause can be represented simply by:

LE: The Schedule specifies that Automatic Early Termination applies to Party A.

The second problem is more serious. It is a special case of the well-known problem of determining the intended meaning of a prepositional phrase, such as "with a telescope" in a sentence, such as "the boy sees the girl with a telescope". In the case of Clause 9, the problem is to determine the meaning of the prepositional phrase "**with respect to any Insured Transaction**". There are at least two interpretations. The phrase can modify either:

1. an Additional Termination Event set forth in Part 1(j)(ix)(a) of this Schedule occurs; or
2. the "Automatic Early Termination" provision of Section 6(a).

In the first interpretation, in relation to Party B, the entire Clause can be represented by:

LE: The Schedule specifies that Automatic Early Termination applies to Party B for an Event of Default.
if the Event of Default of type Section 5(a) (vii) (1), (3), (5), (6) or, to the extent analogous thereto, (8) occurs for Party B at a time T1
and the Event of Default is governed by a system of law
which does not permit termination to take place at a time T2 **and** T2 is after T1
and with respect to any Insured Transaction an Additional Termination Event set forth in Part 1(j)(ix)(a) of this Schedule occurs at a time T3
and T3 is before T2.

The quantification of the variable "**any** Insured Transaction" is ambiguous. Ignoring the context, it is most natural to interpret "any" here as "some", but this makes no sense at all in the given context. It makes more sense to interpret it as meaning that for every Insured Transaction an additional Termination Event occurs. This can be easily translated into Prolog by using the forall meta-predicate, and will not concern us further here.

The second interpretation is much more problematic:

LE: The Schedule specifies that Automatic Early Termination applies to Party B for an Event of Default with respect to a Transaction
if the Event of Default of type Section 5(a) (vii) (1), (3), (5), (6) or, to the extent analogous thereto, (8) occurs for Party B at a time T1
and the Event of Default is governed by a system of law
which does not permit termination to take place at a time T2
and T2 is after T1
and it is not the case that the Transaction is an Insured Transaction.

The Schedule specifies that Automatic Early Termination applies to Party B for an Event of Default with respect to a Transaction
if the Event of Default of type Section 5(a) (vii) (1), (3), (5), (6) or, to the extent analogous thereto, (8) occurs for Party B at a time T1
and the Event of Default is governed by a system of law
which does not permit termination to take place at a time T2
and T2 is after T1
and the Transaction is an Insured Transaction
and an Additional Termination Event set forth in Part 1(j)(ix)(a) of this Schedule has occurred at a time T3
and T3 is before T2.

This interpretation is problematic for two reasons: Firstly, it breaks the template for specifying Automatic Early Termination by adding yet another parameter, which makes it necessary to revise the LE representation of 6(a), to take the additional

parameter into account. Secondly and more seriously from a business modelling perspective, it breaks any sensible model of an ISDA Master Agreement.

At the heart of the ISDA Master Agreement is the concept that the occurrence of an Event of Default such as the bankruptcy of one of the parties, leads to the termination of **all** Transactions governed by the ISDA Master Agreement, upon the Early Termination Date designated by the non-defaulting party. This termination is done by replacing the value of all single Transactions by a single close-out amount, i.e. "close-out netting". Its importance is such that it is often referred to as the key reason parties will use an ISDA Master Agreement to document their derivatives trading relationships.

Following on from this, if such a termination is problematic (and therefore the creation of such a single net close-out amount is problematic) due to the insolvency laws of a jurisdiction, the parties typically agree to use the Automatic Early Termination provision, as it deals with the issue that the applicable insolvency laws may not permit such a close-out netting mechanism to occur once such bankruptcy events occur.

This interpretation of Clause 9 runs the risk of driving a horse and cart through the whole ISDA Master Agreement, essentially carving out the termination of a particular transaction, the Insured Transaction, from the termination of Transactions (and therefore from the single close-out amount) upon the occurrence of an Event of Default such as Bankruptcy.

Although this interpretation of Clause 9 can be represented in LE, the impact is effectively the equivalent of contract heart surgery. We therefore suggest that any representation of the contractual wording of this clause in LE needs to be carefully considered, especially if the representation is to then assist with the onward automation of the performance of contractual obligations. In such a case, It would make sense for the LE representation to be treated as an exception to the general model used for representing contractual obligations; and it would be advisable to manage the representation using appropriate exception management techniques.

8 Related work

As already mentioned, the most closely related work is that on controlled natural language specifications (Fuchs and Schwitter 1995, 1996; Schwitter 2002; Fuchs et al. 2008; Fuchs 2013). In comparison with this work, LE is much closer to logic programming syntax. For example, whereas ACE and PENG use a domain-specific dictionary to identify whether a word is a noun, adjective or verb, LE employs templates, which translate directly into Prolog predicates and arguments.

ACE and PENG are much closer to ordinary, ambiguous English, with conventions for disambiguation, which can sometimes be unintuitive (Fuchs 2018). The basic form of LE avoids these ambiguities because it is simply an alternative syntax for an extended version of LP. Nonetheless, to a large extent, the examples investigated in this paper could equally well be formulated in a restricted version of ACE or PENG.

LE shares with LP the implicit quantification of variables. However, the natural reading of variables in LE extends basic LP with existentially quantified facts and rules. This feature of LE and its implementation by skolemization is shared with ACE and PENG (Fuchs and Schwitter 1995). It is also shared with such other extensions of LP as existential rules (Cali et al. 2010) and ∀∃-rules (Baget et al. 2011).

Although LE can be used as a general-purpose computer language, in this paper we have focussed on its application to the standardisation of legal language, inspired by similarities between the syntax of rules in LP and the style of well-written legislation (Kowalski 1992). As a consequence, LE can be viewed as a domain-specific language (DSL) for legal applications. Viewed in this way, it bears comparison with other English-like DSLs, such as Blawx (Morris 2020, 2021), Lexon (Diedrich 2020), Oracle Intelligent Advisor (Lee 2020) and SBVR Structured English (SE) (OMG 2015). A recent survey and comparison of languages for smart and computable contracts is given in (Clack 2021).

Blawx is a combination of the logic programming language Flora-2 and the visual coding environment Blockly, a descendant of Scratch developed for teaching children to code. Lexon on the other hand combines syntactic sugar for logic programs with higher-order logic, and compiles into Solidity, the programming language developed for the Ethereum blockchain. Idelberger (2020) compares the application of Solidity, Prolog and Lexon to a software licensing contract, and concludes that Lexon is the most promising candidate for such applications.

The relationship with Oracle Intelligent Advisor is more difficult to ascertain, because of its commercial orientation and its lack of scholarly publications. It was originally based on production rule technology, but seems to have morphed over the years into a system that uses LP-style rules with an English-like syntax. SBVR SE, on the other hand, provides an English-like syntax for a business rule language based on a combined alethic and deontic modal logic.

An important feature of legal language is its use of rules and exceptions. In this paper we have expressed rules and exceptions in a basic form of LE, which maps directly into LP. However, it is possible to extend LE, to represent rules and exceptions in the more common form, copying the extensions of LP, which transform the common form into the basic form (Kowalski and Sadri 1991; Sato et al. 2010). Similar transformations have been used to translate defeasible logic, which represents rules and exceptions in a common form, into LP (Antoniou et al. 2006).

One of the main criticisms of LP applied to legal applications is that it does not directly support the representation of deontic modalities such as obligation, prohibition and permission. For example, (Antoniou et al. 2021) argue that, in the case of ASP, the number of rules needed to define deontic predicates in LP can be excessively large. This contradicts our own experience, which suggests that deontic predicates can be defined compactly and perspicuously in LP form, as exemplified in Section 6 by the rule relating permission to designate an event to the actual occurrence of the event.

However, the LE representation of deontic modalities, using such phrases as *it is permitted that* and *it is obligatory that* rises above such concerns about the underlying logic of modality. It is compatible both with the use of modal logic and with

the use of deontic predicates. As a consequence, it has the potential to serve as an English syntax for a higher-level rule interchange language in the spirit of the Legal Knowledge Interchange Format (Gordon et al. 2009) and LegalRuleML (Athan et al. 2013).

LE is inspired in part, not only by the style of well written legal language, but also by more general guidelines (Williams 1990; Williams and Bizup 2017) for writing clear and simple English. It follows the guiding principle for writing clearly that the closer the syntax of a sentence is to the meaning of the sentence, the easier it is for a human or machine to understand and process the sentence.

Although LE is computer-processable, it can also be used just to clarify and simplify human communications, similarly to the way that Simplified Technical English (STE) is used for technical documentation in the aerospace and defence industry (ASD-STE100 2017). LE shares with STE such characteristics as the use of "one word for one meaning and one part of speech for one word". LE also shares with STE that it "was created for the maximum benefit of the reader. This does not necessarily mean it is simple to write." Along with other requirements, to write LE or STE correctly, you must "be able to structure thoughts and ideas logically and understand your reader's needs".

9 Conclusions

We have shown that at the topmost level the entire sample of representative AET Schedule clauses can be expressed in LE, and we have suggested that in some cases the LE representations may be easier to understand than the original English clauses. Moreover, the resulting LE representations are computer intelligible, whereas the original English is not.

On the other hand, the process of reformulating the original English clauses in LE proved to be a significant challenge. Some of the difficulty resulted from the inefficiency of the two phase approach, in which the first author first analysed the logic of the English, and the second author later refined the analysis and drew attention to errors. It would have been more efficient if the two authors had worked together from the outset, as in the standard methodology whereby a knowledge engineer and a domain expert develop a computer implementation, working together as a team (which was not possible in this case, because of timing constraints).

Other difficulties were due to the complexities of the original English, as well as to the need to choose between alternative ways of reformulating the same English clauses. In general, there is no unique or best way to express a text in LE, in the same way that there is no unique or best way to write an English text or a computer program. Moreover, there is no significant corpus of LE examples to serve as a guide to writing style.

Nonetheless, as Genesereth (2015) has claimed "the most popular approach to building Computational Law systems today is based on Computational Logic", which is logic programming by another name. This widespread use of LP for building legal applications suggests a two-pronged, bidirectional strategy for implementing legal texts in LE: working directly from the legal text into LE, as we have done

in this paper, and working indirectly from the legal text into an LP language such as Prolog or ASP, followed by translation from the LP language into LE. Each direction can provide guidance for the other. Moreover, no matter whether the implementation in LE proceeds directly from the original legal text or indirectly via an LP language, the implementation can be facilitated by providing writers with the kinds of authoring tools that come with related systems such as ACE, PENG, Blaux, Lexon and Oracle Intelligent Advisor.

Another possibility is to use natural language processing (NLP) tools to help identify the underlying logic of the legal text. For example, Wyner (2015) investigated the use of the open source tool C&C/Boxer (Bos 2008), which parses and semantically represents text, whereas Holzenberger et al. (2020) applied machine reading tools, including BERT (Devlin et al. 2018), to a large corpus of US tax law documents. Both studies showed that there is a large gap between the meanings generated by the NLP tools and the intended meanings represented in Prolog. However, it may be possible to reduce the gap by interposing a controlled natural language like LE between the unrestricted legal text and the LP representation.

In any case, whether the translation from a legal English text to its LE representation is performed purely manually or with the aid of NLP tools, at this stage of development, the translation process is undoubtedly challenging. Moreover, the skills taught in schools today for writing imperative programs can interfere with the skills needed for writing logic programs. The logic and natural language writing skills, which would help with writing Logical English, have virtually no place in the educational curriculum. It would be ironic if, one day, the situation becomes reversed, and some kind of Logical English is taught as a computer language for children (Kowalski 1982), helping to introduce children to logic and writing skills through the back door.

Acknowledgements The first author is grateful to D2LT for the initial support for this work. Many thanks also to Miguel Calejo, Stuart Camplin, Norbert Fuchs, Jeffrey Golden, Robert Peat and Luis Pereira for comments on an earlier draft of the paper. Special thanks to Christopher Clack, Jacinto Davila and Rolf Schwitter for their detailed comments, which led to more extensive revisions of the paper.

Funding Akber Datoo was supported fully and Robert Kowalski partly by D2 Legal Technology.

Declaration

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Antoniou G, Billington D, Governatori G, Maher MJ (2006) Embedding defeasible logic into logic programming. Theory Pract Logic Progr 6(6):703–735

ASD-STE100 (2017) ASD simplified technical english. http://www.asd-ste100.org/

Antoniou G, Baryannis G, Batsakis S, Governatori G, Islam MB, Liu Q, Robaldo L, Siragusa G (2021) Large-scale legal reasoning with rules and databases. IfCoLoG J Logic Appl 8(4):911–939

Athan T, Boley H, Governatori G, Palmirani M, Paschke A, and Wyner A (2013) Oasis LegalRuleMl. In proceedings of the fourteenth international conference on artificial intelligence and law pp. 3–12

Baget JF, Leclère M, Mugnier ML, Salvat E (2011) On rules with existential variables: walking the decidability line. Artif Intell 175(9–10):1620–1654

Barklund J, Hamfeld A (1994) Hierarchical representation of legal knowledge with metaprogramming in logic. J Logic Program 18:55–80

Bench-Capon TJ, Robinson GO, Routen TW, and Sergot MJ (1987) Logic programming for large scale applications in law: a formalisation of supplementary benefit legislation. In proceedings of the 1st international conference on artificial intelligence and law pp. 190–198

Bos J (2008) Wide-coverage semantic analysis with boxer. In Bos J, and Delmonte R (eds) Semantics in text processing. STEP 2008 conference proceedings, research in computational semantics, pp. 277–286. College Publications

Calì A, Gottlob G, Lukasiewicz T, Marnette B, and Pieris A (2010) Datalog+/-: A family of logical knowledge representation and query languages for new applications. In 2010 25th annual IEEE symposium on logic in computer science pp. 228–242. IEEE

Choi SJ, Gulati GM (2005) Contract as statute. Mich l Rev 104:1129

Clack CD (2021) Languages for smart and computable contracts. https://arxiv.org/abs/2104.03764

Datoo A (2019) Legal data for banking: business optimisation and regulatory compliance. John Wiley & Sons

Datoo A, and Clack CD (2021) Smart close-out netting. J Secur Oper 13–To appear in July 2021

Davila J (2017) Rock, paper, scissors. http://demo.logicalcontracts.com/example/RockPaperScissorsBaseEN.pl

Devlin J, Chang MW, Lee K, and Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805

Diedrich H (2020) Lexon bible: hitchhiker's guide to digital contracts. Wildfire Publishing

Fu Z (2020) Logical english (LE) for representing legal documents, MSc thesis. Imperial college London

Fuchs NE (2013) Attempto project. http://attempto.ifi.uzh.ch/site/

Fuchs NE (2018) Understanding texts in attempto controlled english. CNL. pp 75–84

Fuchs NE, and Schwitter R (1995) Specifying logic programs in controlled natural language. In CLNLP 95, workshop on computational logic for natural language processing, Edinburgh

Fuchs NE, and Schwitter R (1996) Attempto controlled english (ace). In EMISA workshop natürlichsprachlicher entwurf von informationssystemen–grundlagen, methoden, Werkzeuge, Anwendungen, Ev. Akademie Tutzing

Fuchs NE, Kaljurand K, Kuhn T (2008) Attempto controlled english for knowledge representation. Reasoning web. Springer, Berlin, Heidelberg, pp 104–124

Genesereth M (2015) Computational law: the cop in the backseat. In third annual futurelaw conference

Gordon TF, Governatori G, Rotolo A (2009) Rules and norms: requirements for rule interchange languages in the legal domain. In: Governatori G, Hall J, Paschke A (eds) Rule representation, interchange and reasoning on the web. Springer, pp 282–296

Guy SC, Schwitter R (2017) The PENG ASP system: architecture, language and authoring tool. Lang Resour Eval 51(1):67–92

Holzenberger N, Blair-Stanek A, and Van Durme B (2020) A dataset for statutory reasoning in tax law entailment and question answering. arXiv Preprint arXiv:2005.05257

Idelberger F (2020) Merging traditional contracts (or law) and (smart) e-contracts–a novel approach. MLR 2020. The 1st workshop on models of legal reasoning

ISDA (2020) What is the ISDA clause library? https://www.isda.org/a/DZdEE/ISDA-Clause-Library-Project-Memo.pdf

Karadotchev V (2019) First steps towards logical english. MSc thesis. Imperial college London

Kowalski R (1982) Logic as a computer language for children. In proceedings of European conference on artificial intelligence. Orsay, France

Kowalski R (1990) English as a logic programming language. N Gener Comput 8(2):91–93
Kowalski R (1992) Legislation as logic programs. In: Comyn G, Fuchs NE, Ratcliffe MJ (eds) Logic programming in action. Springer, Verlag, pp 203–230
Kowalski R (2019) Logical english. http://www.doc.ic.ac.uk/~rak/papers/Logical%20English.pdf
Kowalski R (2020) Logical english, logic and practice of programming (LPOP). http://www.doc.ic.ac.uk/~rak/papers/LPOP.pdf
Kowalski RA, Sadri F (1991) Logic programs with exceptions. N Gener Comput 9(3–4):387–400
Kowalski R, Sadri F (2015) Reactive computing as model generation. N Gener Comput 33(1):33–67
Kowalski R, Sadri F (2016) Programming in logic without logic programming. Theory Pract Logic Program 16(03):269–295
Kowalski R, Sergot M (1986) A logic-based calculus of events. New Gener Comput 4(1):67–95
Kuhn T (2014) A survey and classification of controlled natural languages. Comput Linguist 40(1):121–170
Lee J (2020) Oracle intelligent advisor—best practice guide for policy modelers. https://www.oracle.com/technetwork/apps-tech/policy-automation/learnmore/opabestpracticeguidev12-3697709.pdf
Lokhorst GJC (1996) Reasoning about actions and obligations in first-order logic. Stud Logica 57(1):221–237
Marín RH, and Sartor G (1999) Time and norms: a formalisation in the event-calculus. In proceedings of the 7th international conference on artificial intelligence and law. pp. 90–99
Morris J (2020) Blawx alpha: user friendly rules as code on the web. https://www.blawx.com/
Morris J (2021) Rules as code: how technology may change the language in which legislation is written, and what it might mean for lawyers of tomorrow. TECHSHOW 2021. https://s3.amazonaws.com/us.inevent.files.general/6773/68248/1ac865f1698619047027fd22eddbba6e057e990e.pdf
OMG (2015) Semantics of business vocabulary and business rules (SBVR), v1.3. https://www.omg.org/spec/SBVR/1.3/PDF
Prakken H, Sartor G (2015) Law and logic: a review from an argumentation perspective. Artif Intell 227:214–245
Reiter R (1989) Towards a logical reconstruction of relational database theory. Readings in artificial intelligence and databases Morgan Kaufmann. Elsevier, pp 301–327
Satoh K, Asai K, Kogawa T, Kubota M, Nakamura M, Nishigai Y, Shirakawa K, Takano C (2010) PROLEG: an implementation of the presupposed ultimate fact theory of Japanese civil code by PROLOG technology. JSAI international symposium on artificial intelligence. Springer, Berlin, Heidelberg, pp 153–164
Schwitter R (2002) English as a formal specification language. In proceedings 13th international workshop on database and expert systems applications. pp. 228–232. IEEE
Sergot MJ (1982) Prospects for representing the law as logic programs. Academic Press, Logic programming, pp 33–42
Sergot MJ, Kamble AS, and Bajaj KK (1991) Indian central civil service pension rules: a case study in logic programming applied to regulations. In proceedings of the 3rd international conference on artificial intelligence and law. pp. 118–127
Sergot MJ, Sadri F, Kowalski RA, Kriwaczek F, Hammond P, Cory HT (1986) The British nationality act as a logic program. Commun ACM 29(5):370–386
Stenning K, Van Lambalgen M (2012) Human reasoning and cognitive science. MIT Press
Vranas PB (2008) New foundations for imperative logic I: logical connectives, consistency, and quantifiers. Noûs 42(4):529–572
Wielemaker J, Riguzzi F, Kowalski RA, Lager T, Sadri F, Calejo M (2019) Using SWISH to realize interactive web-based tutorials for logic-based languages. Theory Pract Logic Program 19(2):229–261
Williams J (1990) Toward clarity and grace. The University of Chicago, Chicago
Williams J, Bizup J (2017) Lessons in clarity and grace, Twelfth. Pearson
Wyner A (2015) From the language of legislation to executable logic programs. Logic in the theory and practice of lawmaking. Springer, Cham, pp 409–434

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.