Logical English for Law

Robert Kowalski[1], Jacinto Dávila[2,3], Galileo Sartor[4], and Miguel Calejo[5]

[1] Department of Computing, Imperial College, London, UK
[2] Contratos Lógicos. C.A.
[3] Universidad de Los Andes Mérida, Venezuela
[4] Department of Computing, University of Turin, Turin
[5] Logical Contracts, Lisbon, Portugal

Abstract. In this paper we summarise the key features of Logical English (LE) as syntactic sugar for logic programming languages such as pure Prolog, ASP and s(CASP); and we illustrate LE with examples from the Italian citizenship legislation and the US Tax Code.

Keywords: Logic Programming, Prolog, Controlled Natural Language, Legal Rule Modelling, Logical English

1 Introduction

Logical English (LE) exploits the unique feature of Prolog-like logic programming (LP), that LP is the only programming paradigm based on the use of logic for human thinking and communication. By exploiting this feature, LE becomes a general-purpose programming language, which can be understood with only a reading knowledge of English and without any technical training in computing, mathematics or logic.

LE is not only a Turing-complete computer programming language. It has the potential to represent and reason with a broad range of human knowledge, as shown by its ability to codify the language of law.

Consider the following example, written in LE, and its translation into Prolog.

Ordinary English:

> *All meetings with unvaccinated people are prohibited unless they are excused.*

Logical English:[6]

> *A meeting is prohibited*
> *if a person attends the meeting*
> *and the person is unvaccinated*
> *and it is not the case that the meeting is excused.*

Prolog:

is_prohibited(A) :-
attends(B, A), is_unvaccinated(B), not is_excused(A).

The implementation of LE in SWISH[8] translates LE programs and queries into Prolog, uses Prolog to answer queries, and translates answers and explanations into LE English syntax.

The example illustrates some of the following characteristics of LE:

- LE avoids pronouns, which are a major source of ambiguity, as in the case of "they", which in this example could refer either to meetings or to people.
- LE represents variables by common nouns prefixed by a determiner such as "a", "an" or "the". The indefinite determiner, "a" or "an", introduces the first occurrence of a variable in a rule. The definite determiner, "the" is used for all later occurrences of the same variable in the same rule. As in Prolog (with some exceptions), all variables are implicitly universally quantified with scope being the rule in which they occur. This means that variables in different rules have no relationship with one another.
- Sentences in LE are either facts, or rules, as in Prolog. Rules have the Prolog-like conditional form *conclusion if conditions*, where the *conclusion* is an atomic sentence and the *conditions* are a combination of atomic sentences, typically connected by *and*. But *conditions* can also be connected by *or* and negation, written in the form *it is not the case that.* The relative precedence of the logical connectives is indicated by indentation (not illustrated in this example).
- As a matter of style and in the interests of greater precision, common nouns are preferably expressed in the singular, and verbs are expressed in the present tense. The temporal relationship between events and time-varying facts can be expressed, if necessary, by referring to time explicitly.

LE inherits the feature of Prolog that propositions can occur as arguments of higher-order or meta-level predicates. LE uses this to represent deontic modalities (obligation, prohibition, permission) and other propositional attitudes (notification, belief, desire, dislike). For example, here the keyword *that* introduces the proposition *a meeting is prohibited at a time T1* as an argument of the meta-predicate *the person is notified*:

a person violates the rules at a time T2
if the person is notified that a meeting is prohibited at a time T1
and the person attends the meeting at T2
and T1 is before or at the same time as T2.

As this rule also shows, a variable can be given a symbolic name.

Atomic sentences, which are facts, the conclusions of rules, or constituents of the conditions of rules, are instances of predicates declared by means of templates, such as:

**a person* violates the rules at *a time*,*
**a person* is notified that * message*,*
**an eventuality* is prohibited at *a time*.*

where the asterisks identify the arguments of the predicates.

We have used the implementation of LE in SWISH to represent a wide range of legal texts, helping to identify ambiguities, to explore modifications and alternative representations of the same text, and to compare the logical consequences of the alternatives.

2 The Italian Citizenship Example

We are also developing analogues of LE for other natural languages, such as Spanish and Italian. Figure 1 shows both an LE representation and a corresponding LI representation of Article 1 of Act No. 91 of 5 February 1992:

> 1. E' cittadino per nascita: a) il figlio di padre o di madre cittadini; b) chi e' nato nel territorio della Repubblica se entrambi i genitori sono ignoti o apolidi, ovvero se il figlio non segue la cittadinanza dei genitori secondo la legge dello Stato al quale questi appartengono.

Google translate gives the following translation into English:

> Citizen by birth: a) the child of a citizen father or mother; b) who was born in the territory of the Republic if both parents are unknown or stateless, or if the child does not follow the citizenship of the parents according to the law of the state to which these belong.

Here both the English condition "the child does not follow the citizenship of the parents according to the law of the state to which these belong" and its Italian counterpart, taken literally, seem to cover only the case where both parents have the same citizenship. Moreover, both the Italian "ovvero se" and the corresponding English "or if" seem to relate to a separate alternative from the alternatives that precede it. These readings of the natural language texts leave uncovered such deserving cases as the child having one parent who is stateless or unknown, and another parent who cannot pass on its citizenship(s) to its child. It seems doubtful that this would have been the intention of the law.

The LE and LI representations in figure 1 1.1 [6] incorporate one intended interpretation of Article 1.1. Of course, other interpretations are possible, and they could also be represented in LE.

Figure 1 also illustrates two further features of LE: the use of indentation to represent the relative strength of binding of the logical connectives, and the LE construction "for all cases in which ... it is the case that ... ", which translates into "forall" in Prolog.

[6] https://logicalenglish.logicalcontracts.com/p/italian_citizen_new.pl
https://logicalenglish.logicalcontracts.com/p/cittadinanza_italiana.pl

```
the knowledge base italian_citizen_new includes:

a person A is an italian citizen
if the person A is an italian citizen by birth.

a person A is an italian citizen by birth
if a person B is the parent of A
and B is an italian citizen.

a person A is the parent of a person B
if A is the father of B
    or A is the mother of B.

a person A is an italian citizen by birth
if A is born in italy
and for all cases in which
    a person B is the parent of A
    it is the case that
    B is stateless
        or B is unknown
        or A does not follow the citizenship of B.
```

```
la base di conoscenza cittadinanza_italiana include:

una persona A ha la cittadinanza italiana
se A ha la cittadinanza italiana per nascita.

una persona A ha la cittadinanza italiana per nascita
se una persona B è genitore di A
e B ha la cittadinanza italiana.

una persona A è genitore di una persona B
se A è madre di B
    o A è padre di B.

una persona A ha la cittadinanza italiana
se A è nato in italia
e per tutti i casi in cui
    una persona B è genitore di A
    è provato che
    B è sconosciuto/a
        o B è apolide
        o A non segue la cittadinanza di B.
```

Fig. 1. The Italian Citizenship Example

In this representation of Italian citizenship, the possibility that a parent is unknown is left explicit, to more closely reflect the wording of the original legal source. This possibility could also be expressed implicitly, using negation as failure, to conclude that a parent of a person is unknown if information about that parent is missing from the knowledge base or from the scenario. It is in fact possible, with the current representation, to say that a person is born in italy, and not to give any information about the parents at all. The "forall" condition would be satisfied (vacuously), and the person would be granted Italian citizenship.

The presence or absence of a fact that a person A is the parent of a person B may depend on such circumstances of the birth as whether the parent A (mother) decides to be recognized as a parent if B, or instead decides to abandon the child B. Both possibilities, that a parent is unknown (implicitly through negation as failure or explicitly by means of an fact that the parent is unknown), are compatible with the legal source. Moreover, the ability to represent both possibilities in LE may help to remove the ambiguity of the rule and it assist in its automation.

3 A Tax Law Example

LE and its counterparts for other natural languages can be used to codify legal rules to support their complete or partial automation. But they can also be used to assist with the drafting of legal rules, to help ensure that the rules actually express their intended interpretation. Used in this way, for both purposes, LE can provide powerful support for both drafting and applying the law, as envisaged in the campaign to represent Rules as Code (RAC) [11].

Within this RAC context, we have investigated §121 of the US Internal Revenue Code, following the lead of the Catala project [9]. Figure 2 shows an LE representation of a portion of §121, which deals with the exclusion of gain from the sale of a principal residence:

> (Subsection a) Exclusion
> Gross income shall not include gain from the sale or exchange of property if, during the 5-year period ending on the date of the sale or exchange, such property has been owned and used by the taxpayer as the taxpayer's principal residence for periods aggregating 2 years or more.

Notice that the condition of the sentence is ambiguous.

- It could mean that the periods during which the taypayer both owns and uses the property within the 5-year period aggregate to 2 years.
- Or it could mean that the periods during which the taypayer owns the property within the 5-year period aggregate to 2 years, and the periods during which the taypayer uses the property within the 5-year period aggregate to 2 years.

The authors of [9] do not mention that the sentence is ambiguous. However, the LE representation in figure 2 follows the Catala implementation of §121, which assumes that the drafters of the Tax Code intended the second interpretation[7]. Of course, the alternative interpretation could also be represented in both LE and Catala.

Subsection (b)1 of §121 defines a cap of $ 250,000 on the amount of gain that can be excluded from a sale or exchange of property. But the Code itself does not express the common sense understanding of the cap as limiting the amount that can be excluded. The Caltala implementation builds this understanding into the representation of subsection (a) itself. However, the LE implementation expresses this understanding separately in lines 126-132 of figure 2.

As is well-documented in the field of AI and law, a typical legal document consists of rules and exceptions, as well as exceptions to exceptions, etc. In this regard, the US Tax Code is not exceptional. In particular, subsection (a) is subject to the exception:

> (3) Application to only 1 sale or exchange every 2 years
> (A) In general
> Subsection (a) shall not apply to any sale or exchange by the taxpayer if, during the 2- year period ending on the date of such sale or exchange, there was any other sale or exchange by the taxpayer to which subsection (a) applied.

To represent this exception, we add an extra, explicit condition (in lines 45-46 of figure 2) to the rule for subsection (a), expressing that it is not the case that

[7] https://catala-lang.org/en/examples/us-tax-code

the exception holds. We also interpose an intermediate conclusion (on line 42) and an intermediate condition (on line 40) in the representation of subsection (a), expressing explicitly that subsection (a) applies. The exception itself is on lines 118-124.

In general, rules and exceptions are represented in Prolog and most other logic programming languages by adding such otherwise implicit conditions (that the contrary of the conclusion does not hold) explicitly. However, Satoh [10] has argued that rules written in this Prolog form are hard for lawyers to understand. It is easier for lawyers to understand rules written more simply with unwritten implicit conditions, as in the Prolog-based Legal reasoning support system, Pro-leg [10]. We agree with this approach, and plan to incorporate such implicit conditions in a future version of LE.

Figure 3 shows an example scenario and query, both written in LE syntax. The SWISH implementation of LE allows several such scenarios and several such queries in the same document. Figure 4 shows the answer and proof tree obtained by combining scenario one with query one. The proof tree is an explanation of the answer to the query, given the scenario and the more general rules in the knowledge base. The last part of the proof, highlighted in red, displays the conditions that could not be proved, justifying the conclusion that the exception does not hold in the given scenario.

4 Conclusions

Our experience with using LE for many practical, proof-of-concept applications suggests that LE has many valuable applications, which are not restricted to the automation of legal rules. These applications include the disambiguation of legal rules written in natural language, as well as the exploration of the logical consequences of the rules, in the context of different scenarios.

These applications are facilitated by the fact that users can read, understand and use LE without any technical training in logic, computing or mathematics. But, although LE may be easy to read, at this stage in its development, it is not easy to write. The fact that the drafters of the Italian citizenship law and of the US Tax Code did not identify the ambiguities in their legal texts proves how difficult it can be to express information clearly and unambiguously in natural language.

Even if LE were never used to help automate the application of legal rules, it would serve a useful purpose as a discipline in training writers to express themselves in terms that readers can more readily understand.

```
gross income of a taxpayer excludes gain from a sale or exchange of a property at a date if
    subsection (a) applies to the sale or exchange of the property by the taxpayer at the date.

subsection (a) applies to a sale or exchange of a property by a taxpayer at a date if
    the taxpayer meets the ownership requirements of subsection (a) with respect to the sale or exchange of the property at the date
    and the taxpayer meets the use requirements of subsection (a) with respect to the sale or exchange of the property at the date
    and it is not the case that
        subsection (a) shall not apply to the sale or exchange of the property by the taxpayer.

a taxpayer meets the ownership requirements of subsection (a) with respect to a sale or exchange of a property at a date
    if the sale or exchange of the property occurs at the date
    and a period of 5 years ends at the date
    and the property has been owned by the taxpayer for periods aggregating 2 years or more during the period.

a taxpayer meets the use requirements of subsection (a) with respect to a sale or exchange of a property at a date
    if the sale or exchange of the property occurs at the date
    and a period of 5 years ends at the date
    and the property has been used by the taxpayer as principal residence for periods aggregating 2 years or more during the period.

subsection (a) shall not apply to a sale or exchange of a property by a taxpayer
    if the sale or exchange of the property occurs at a date
    and an other sale or exchange of a second or the same property occurs at a second date
    and the other sale or exchange is different from the sale or exchange
    and a period of 2 years ends at the date
    and the second date is included in the period
    and subsection (a) applies to the other sale or exchange of the second or the same property by the taxpayer at the second date.

the amount of gain to be excluded for a taxpayer from a sale or exchange under subsection (a) is an amount G
    if the received income for the sale or exchange is an amount I
    and the amount of gain excluded for the taxpayer from the sale or exchange under subsection (a) shall not exceed a cap
    and     the cap >= I
            and G is I
        or  the cap < I
            and G is the cap.
```

Fig. 2. §121 of the US Internal Revenue Code subsection (a)

```
scenario one is:
    the sale of the house occurs at 2022-06-20.
    the received income for the sale is 300000.
    the given period of 5 years ends at 2022-06-20.
    first set of periods of the taxpayer owing the house aggregates to 2 of years.
    first set of periods is contained in the given period.
    second set of periods of the taxpayer using the house as principal residence aggregates to 3 of years.
    second set of periods is contained in the given period.

scenario two is: …
scenario three is: …
query one is:
    gross income of which taxpayer excludes gain from which exchange of which property at which date.
```

Fig. 3. A scenario and query for subsection (a)

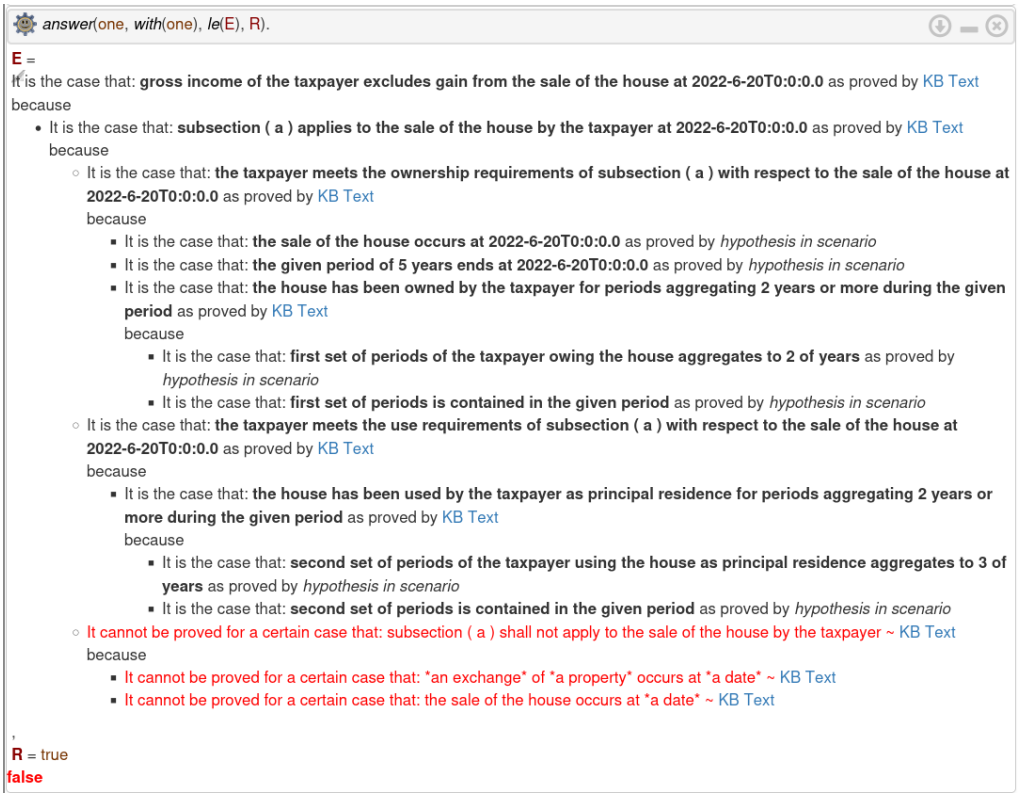

Fig. 4. Explanation of the answer of a query in LE

References

1. Kowalski R (1990) English as a logic programming language. New Generation Comput 8(2):91–9.
2. Kowalski R (1992) Legislation as Logic Programs In: Logic Programming in Action (eds. G. Comyn , N. E. Fuchs, M. J. Ratcliffe), Springer Verlag, 203-230.
3. Kowalski R (2020) Logical English, In Proceedings of Logic and Practice of Programming (LPOP).
4. Kowalski, R. and Datoo, A., 2022. Logical English meets legal English for swaps and derivatives. Artificial Intelligence and Law, 30(2), pp.163-197.
5. Kowalski, B., Dávila, J. and Calejo, M., Logical English for legal applications. XAIF, Virtual Workshop on Explainable AI in Finance (2021).
6. LE unvaccinated: https://le.logicalcontracts.com/p/unvaccinatedupdate.pl, last accessed 2022/09/06.
7. Sartor, G., Dávila, J., Billi, M., Contissa, G., Pisano, G. and Kowalski, R., 2022. Integration of Logical English and s (CASP), 2nd Workshop on Goal-directed Execution of Answer Set Programs (GDE'22), August 1, 2022.
8. Jan Wielemaker, Torbjörn Lager , and Fabrizio Riguzzi, 2015. SWISH: SWI-Prolog for sharing. arXiv preprint arXiv:1511.00915.
9. Huttner, L., Merigoux, D. & Chataing, N. Catala: Moving Towards the Future of Legal Expert Systems. (2022), https://hal.inria.fr/hal-02936606
10. Satoh, K., Asai, K., Kogawa, T., Kubota, M., Nakamura, M., Nishigai, Y., Shirakawa, K. and Takano, C., 2010, November. PROLEG: an implementation of the presupposed ultimate fact theory of Japanese civil code by PROLOG technology. In JSAI international symposium on artificial intelligence (pp. 153-164). Springer, Berlin, Heidelberg.
11. Mohun, James, and Alex Roberts. "Cracking the code: Rulemaking for humans and machines." (2020).