# Logical English

=

Logic + English + Computing

Bob Kowalski
Imperial College London

LPOP 2020
15 November 2020

# Logical English as a Computer Language

A general purpose logic and computer language,
based on logic programming.

Destructive change of state, as in the real world.

Applications to the automation of
legal regulations and smart contracts.

Readable without training in mathematics, computing or logic.

Easier to write than conventional logic programs?

# A short, personal history of Logical English

I failed English writing skills at the U of Chicago (1958).

Ambiguous pronouns.
Sentences that sounded good, but had no meaning.

Logic as a Computer Language for Children (1978+).

The British Nationality Act as a Logic Program (1985-86).

Writers' Workshops at Imperial and Japan (~1998-2015).

Logical Contracts and Logical English (2017-2020).

# Relationships with other work

Controlled natural languages, implemented in Prolog, for general knowledge representation:

- Attempto Controlled English (ACE) [Fuchs and Schwitter, 1996; Fuchs et al, 2008; Fuchs, 2013]
- PENG [Schwitter, 2002]

English-like domain-specific languages for legal applications, based on logic programming

- Blaux [Morris, 2020]
- Lexon [Diedrich, 2020]

Syntactic sugar for the logic programming language ASP + Event Calculus

- $\text{PENG}^{\text{ASP}}$ [Guy and Schwitter, 2017]

SBVR (Semantics of Business Vocabulary and Rules) based on modal logic

- OMG

## The basic form of LE is a sugared syntax for logic programs.

Instead of writing symbolic expressions such as:

$$\forall\ X, Y, A, B1, B2, T1, T2(account\_balance(X, B2, T2) \leftarrow$$
$$(isa(X, account), isa(B2, amount), isa(T2, time),$$
$$transfer(Y, X, A, T1), isa(Y, account), isa(A, amount), isa(T1, time),$$
$$balance(X, B1, T1), isa(B1, amount), isa(T1, time), sum(B1, A, B2), next(T1, T2)))$$

we write “controlled” English expression 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** T1 is the time immediately before T2
**and** the balance in the account is an amount B1 at T1
**and** B2 is the sum of B1 and A.

# Variables in conditions are universally quantified with scope the sentence in which they occur

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 is the sum of B1 and A
**and** T2 is the next time after T1.

Variables are symbols such as X, Y, B1, B2, A, T1, T2,
or signalled by “a”, “an” or “the” before a common noun,
which represents the type of the variable.
“a” and “an” are used for the first occurrence of the variable.
“the” is used for later occurrences of the same variable in the same sentence.

# The plan is to develop LE as a series of extensions, starting from the basic form

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.

To reduce ambiguity LE has no pronouns, such as “he”, “she”, or “it”.
To reduce or eliminate the need for a dictionary,
all nouns and verbs are expressed in the singular.

## Any variable in the conclusion that is not in the conditions is existentially quantified with “wide scope”.

An event of a person acquiring citizenship of the land of oz occurs on a day
**if** the person is born in a place on the day **and** the place is in the land of oz.

A person celebrates the event
**if** the person lives in the land of oz.

This reading of the English article is compatible with the
interpretation of implicit quantifiers in existential (or ∀ ∃ ) rules,
and with the elimination of existential quantifiers by skolemization.

# Implementations

There have been three experimental implementations of variants of LE based on LPS or Prolog, focussed primarily on legal applications

| | |
|---|---|
| Davila, 2017 | rock-paper-scissors |
| Karadotchev, 2019 | ISDA Master Agreements |
| Fu, 2020. | Simplified loan agreement |

# Reactive rules implement observations of external events and execution of internal or external actions.

**If** a player P1 plays a choice C1
**and** another player P2 plays a choice C2
**and** C1 beats C2
**and it is not the case** that the game is over
**then** P1 receives the prize (or give P1 the prize)
**and it becomes the case** that the game is over. (or initiate the game is over)

# Meta-level (or higher-order) logic

A transaction is governed by IsdaAgreement
**if** a confirmation of the transaction states
that the transaction is governed by IsdaAgreement
**and** the transaction commences on a first day
**and** IsdaAgreement is dated as of a second day
**and** the first day is on or after the second day.

# Event calculus ontology without frame axioms

It **becomes the case that** a requirement is defaulted on a day
**when** it is the end of the day
**and t**he requirement is potentially defaulted
**and** the lender delivers a notice to the borrower on another day
**and** the notice is that the requirement is potentially defaulted
**and** the other day is 3 days before the day
**and it is not the case that** the requirement is cured.

# Euclid's Algorithm: Greatest Common Denominator - as presented by Leslie Lamport at https://www.youtube.com/watch?v=wQiWwQcMKuw&feature=youtu.be&ab_channel=NUScast

$$Init_E \ \wedge \ \Box \, Next_E$$

This is Euclid's algorithm written in mathematics.

These are just definitions (abbreviations):

$$Init_E : \quad (x = M) \wedge (y = N)$$

$$\begin{aligned} Next_E : \quad & ( \quad (x > y) \\ & \quad \wedge (x' = x - y) \\ & \quad \wedge (y' = y) \qquad ) \\ & \vee \ ( \quad (y > x) \\ & \quad \wedge (y' = y - x) \\ & \quad \wedge (x' = x) \qquad ) \end{aligned}$$

11

# Euclid's Algorithm: Greatest Common Denominator - in Logical English - not implemented

**It becomes the case that** the value of n is X
**and that** the value of m is Y
**when** requested to find the greatest common divisor of X and Y.

**If** the value of X is V1
**and** the value of Y is V2
**and** V1 is greater than V2
**then** V1 is updated to V1-V2 in the value of X.

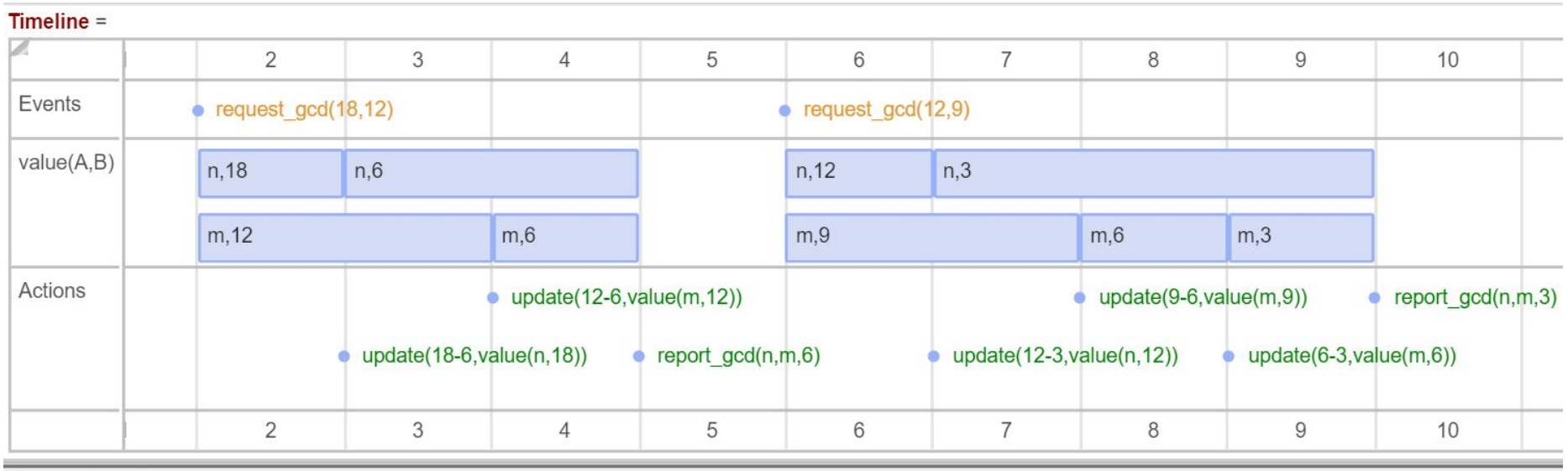
**If** the value of n is V
**and** the value of m is V
**then** the greatest common divisor of n and m is reported as V.

**It becomes not the case that** the value of n is V
**and that** the value of m is V
**when** the greatest common divisor of n and m is reported as V.

# Euclid's Algorithm: Greatest Common Denominator - in LPS online, editable and runnable at http://demo.logicalcontracts.com/p/Euclid's%20Algorithm.pl

```
events request_gcd/2.
fluents value/2.
actions report_gcd/2, assign/2.

request_gcd(X,Y) initiates value(n,X).
request_gcd(X,Y) initiates value(m,Y).

report_gcd(n,m,V) terminates value(n,V).
report_gcd(n,m,V) terminates value(m,V).

if value(X,V1), value(Y,V2), V1 > V2
then V is V1 - V2, update V1 to V in value(X, V1).

if value(n,V), value(m,V)
then  report_gcd(n,m,V).
```

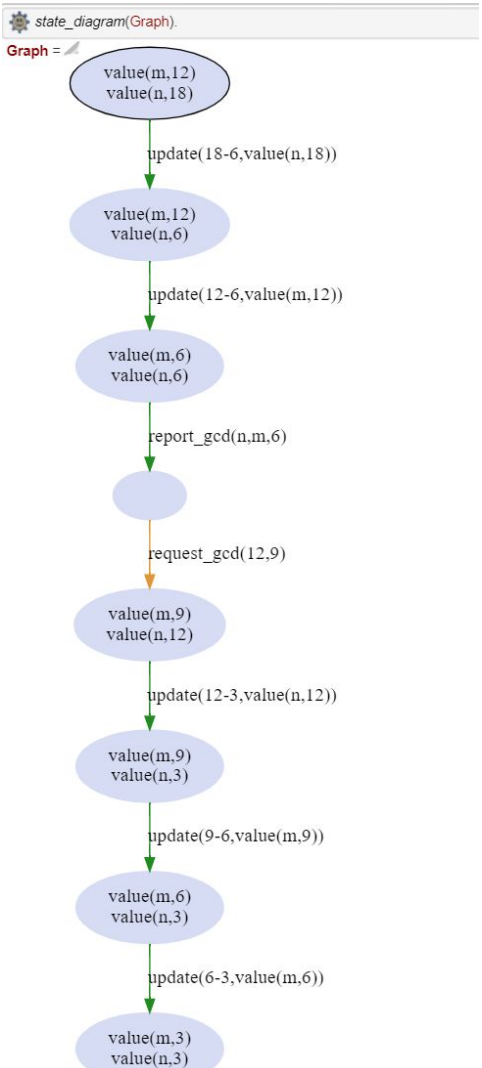
# Euclid's Algorithm: Greatest Common Denominator -
http://demo.logicalcontracts.com/p/Euclid's%20Algorithm.pl




```
?- go(Timeline).
```

# Euclid's Algorithm:
# Greatest Common Denominator -

http://demo.logicalcontracts.com/p/Euclid's%20Algorithm.pl



# Logical English Meets Legal English for Swaps and Derivatives

Robert Kowalski, Imperial College London
Akber Datoo, D2 Legal Technology, London
14 November 2020

## Abstract

In this report, we present an informal introduction to Logical English (LE) in the context of its application to the Automatic Early Termination (AET) clause of ISDA (International Swaps and Derivatives Association) Master Agreements. LE is a controlled natural language, which is designed to be both computer-executable and readable by English speakers without special training. Compared with conventional legal language, LE is designed to be as unambiguous as possible without the loss of expressive power. Compared with conventional computer languages, LE is executed by translating it into a logic programming language such as Prolog, Datalog or Answer Set Programming (ASP).

## Introduction

Logical English (LE) [Kowalski, 1990, 2019] is a work in progress, intended as syntactic sugar for logic programs, which consist of facts, such as “IsdaAgreement is dated as of 03/10/2020”, and conditional sentences (or rules), such as:

> A Transaction is governed by IsdaAgreement
> **if** a confirmation of the Transaction states that the Transaction is governed by IsdaAgreement
> **and** the Transaction commences on a first date
> **and** IsdaAgreement is dated as of a second date
> **and** the first date is on or after the second date.

# Conclusions

- We need to stop teaching children to think like computers.
- We should teach computers to think like humans.
- We should teach children logic and writing skills.
- We should follow the lead of legal scholars:

  Stop using complex language understandable only by technicians.
  Start using plain language understandable by ordinary people.

# ISDA 2002 Master Agreement - Early Termination following Event of Default

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).

# Permission can be represented by meta-order or high-order predicates.

**It is permitted that** a party designates that
an Early Termination Date occurs in respect of every outstanding Transaction at time T3 by performing a notification to another party with effect at time T2 that an Event of Default occurs at time T1
**if** the Event of Default with respect to the other party occurs at time T1
**and** the Event of Default is continuing at time T2
**and it is not the case that**
Automatic Early Termination is specified in the Schedule as applying to the Event of Default and to the other party
and $T2 \leq T3 \leq T2 + 20$ days.

An Early Termination Date occurs in respect of every outstanding Transaction at time T

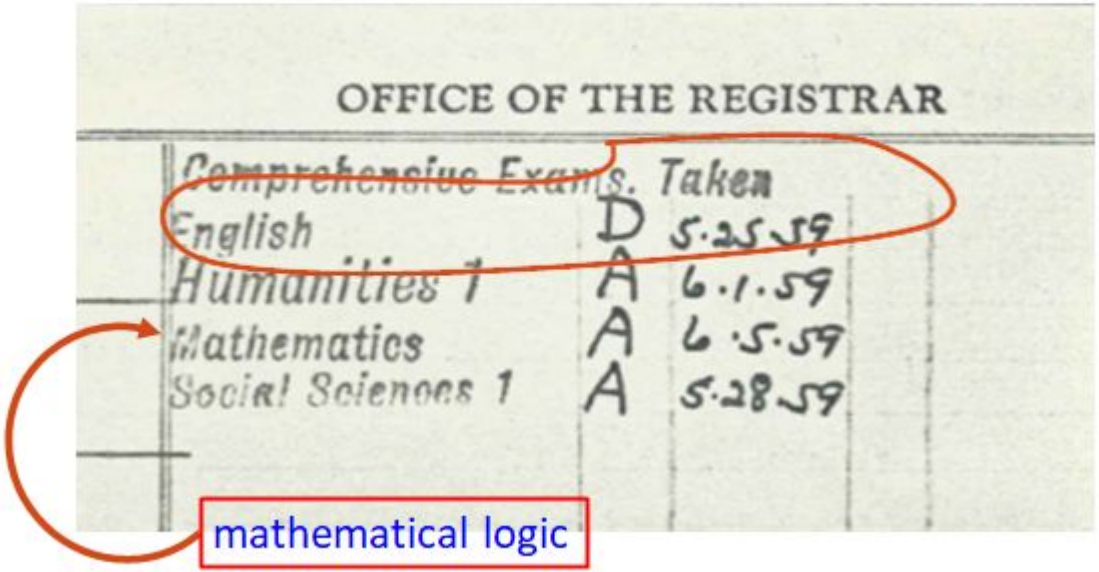
**if** an Event of Default with respect to a party occurs at time T

**and** Automatic Early Termination is specified in the Schedule as applying to the Event of Default and to the party

**and** the Event of Default is specified in Section 5(a)(vii)(1), (3), (5), (6) or, to the extent analogous thereto, (8).

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

# What I learned from the U.C. Core



# What I learned from the University of Chicago Core

"This famed Core curriculum, ...,
is the University of Chicago student's introduction to the tools of inquiry used in every discipline —science, mathematics, humanities, and social sciences."

Kowalski, R. Logic as a Computer Language for Children.
In Proceedings of European Conference on Artificial Intelligence, Orsay, France, July 1982.

John and Bob like Mary

is translated into two atomic sentences

John likes Mary
Bob likes Mary.

But the English sentence

John and Bob like themselves

is ambiguous.

It is a major objective of the project to teach the relationship between natural language syntax and its semantics, where as a first approximation the semantics are expressed in symbolic logic. This is thought to be an important object in its own right as a contribution to the more effective use of natural language: to teach the distinction between English sentences which are clear and precise and English sentences which are imprecise or meaningless.

ELIZABETH II

# British Nationality Act 1981

**1981 CHAPTER 61**

An Act to make fresh provision about citizenship and nationality, and to amend the Immigration Act 1971 as regards the right of abode in the United Kingdom.

[30th October 1981]

BE IT ENACTED by the Queen's most Excellent Majesty, by and with the advice and consent of the Lords Spiritual and Temporal, and Commons, in this present Parliament assembled, and by the authority of the same, as follows:—

## PART I

### BRITISH CITIZENSHIP

*Acquisition after commencement*

Acquisition by birth or adoption.

**1.**—(1) A person born in the United Kingdom after commencement shall be a British citizen if at the time of the birth his father or mother is—

(*a*) a British citizen ; or

(*b*) settled in the United Kingdom.

(2) A new-born infant who, after commencement, is found abandoned in the United Kingdom shall, unless the contrary is shown, be deemed for the purposes of subsection (1)—

(*a*) to have been born in the United Kingdom after commencement ; and

(*b*) to have been born to a parent who at the time of the birth was a British citizen or settled in the United Kingdom.

Sergot, M.J., Sadri, F., Kowalski, R.A., Kriwaczek, F., Hammond, P. and Cory, H.T., 1986. The British Nationality Act as a logic program. *Communications of the ACM*, *29*(5).

**English**

1.-(1) A person born in the United Kingdom after commencement shall be a British citizen
if at the time of the birth his father or mother is
(a) a British citizen; or
(b) settled in the United Kingdom.

**Logic Program**

*X acquires british citizenship by subsection 1.1 at time T*
***if*** *X is born in the uk at time T*
***and*** *T is after commencement*
***and*** *Y is father of X* ***or***
*Y is mother of X*
***and*** *Y is a british citizen at time T* ***or***
*Y is settled in the uk at time T*

Kowalski, R. English as a logic programming language. *New Gener Comput* 8, 91–93 (1990).

New Generation Computing, 8 (1990) 91-93
OHMSHA, LTD. and Springer-Verlag

NEW GENERATION COMPUTING

© OHMSHA, LTD. 1990

**Preface**

# English as a Logic Programming Language

Robert KOWALSKI
*Department of Computing,*
*Imperial College of Science, Technology and Medicine,*
*180 Queen's Gate, London SW7 2BZ, England.*

Ordinary English is generally ambiguous, imprecise and unclear. This makes it less than ideal both for human communication and for communication with computers. To improve the use of English for human communication, English scholars have identified guidelines for good use. These guidelines help to improve clarity by reducing ambiguity, and by reducing the distance between syntactic form and semantic content.

I believe that English formulated in accordance with the guidelines for good use can also serve as a yardstick for evaluating computer languages and knowledge representation formalisms. The language of public notices is especially useful in this respect. It tends to combine the precision of legal language with the clarity and coherence needed to be understood efficiently by ordinary people.

Consider, for example, the English of the following notice posted in the carriages of the London underground.

**Emergencies**

Press the alarm signal button
to alert the driver.

The driver will stop immediately

# Legislation as Logic Programs*

Robert A. Kowalski

Department of Computing
Imperial College of Science, Technology and Medicine
London SW7 2BZ, UK

January 1991
Revised June 1992

**Abstract.** The linguistic style in which legislation is normally written has many similarities with the language of logic programming. However, examples of legal language taken from the British Nationality Act 1981, the University of Michigan lease termination clause, and the London Underground emergency notice suggest several ways in which the basic model of logic programming could usefully be extended. These extensions include the introduction of types, relative clauses, both ordinary negation and negation by failure, integrity constraints, metalevel reasoning and procedural notation.