REASONING OVER LEGAL TEXTS USING LARGE LANGUAGE MODELS AND AUTOMATED REASONING

Verena Praher[1] **Endre Szasz-Revai**[2] **Wolfgang Windsteiger**[3]
[1] RISC Software GmbH [2] Rise2Reality [3] RISC Institute, JKU Linz

July 7, 2026 – SCML-2026, Hagenberg Castle, Austria

JOHANNES KEPLER UNIVERSITY LINZ

INTRODUCTION

Bring computer support to taxation decisions of internationally operating businesses.

First prototype: Transfer Pricing.

Problem:

- When company divisions trade across borders, what prices should they charge?

INTRODUCTION

Bring computer support to taxation decisions of internationally operating businesses.

First prototype: Transfer Pricing.

Problem:

- When company divisions trade across borders, what prices should they charge?
- Prices must comply with rules from many sources: OECD, EU, federal law, and internal guidelines.

INTRODUCTION

Bring computer support to taxation decisions of internationally operating businesses.

First prototype: Transfer Pricing.

Problem:

- When company divisions trade across borders, what prices should they charge?
- Prices must comply with rules from many sources: OECD, EU, federal law, and internal guidelines.
- Deciding which combination of rules applies is non-trivial even for human experts.

INTRODUCTION

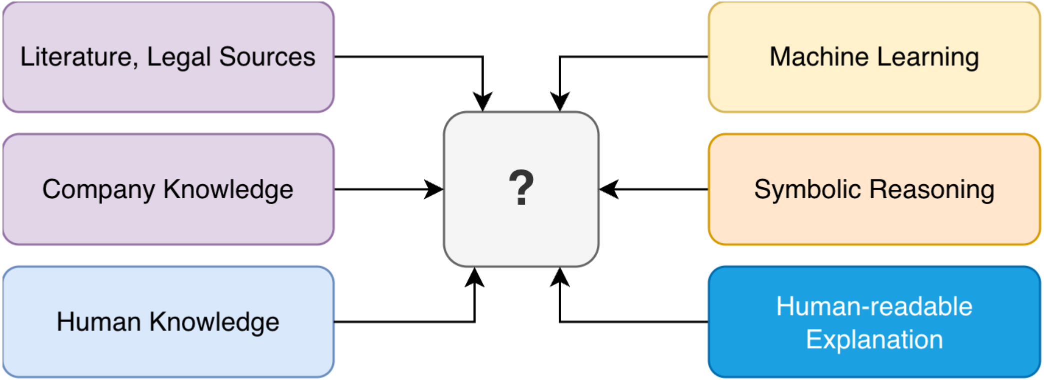
Bring computer support to taxation decisions of internationally operating businesses.

First prototype: Transfer Pricing.

Problem:

- When company divisions trade across borders, what prices should they charge?
- Prices must comply with rules from many sources: OECD, EU, federal law, and internal guidelines.
- Deciding which combination of rules applies is non-trivial even for human experts.
- Decisions must be explainable and traceable back to the rules applied.

PROBLEM SETTING



APPROACH

Combine machine learning with automated reasoning at the problem level.

APPROACH

Combine machine learning with automated reasoning at the problem level.

- No augmentation of methods.

APPROACH

Combine machine learning with automated reasoning at the problem level.

- No augmentation of methods.
  - ML stays ML: translation only; not guided by SC.

APPROACH

Combine machine learning with automated reasoning at the problem level.

- No augmentation of methods.
  - ML stays ML: translation only; not guided by SC.
  - SC stays SC: inference only; not guided by LLM.

APPROACH

Combine machine learning with automated reasoning at the problem level.

- No augmentation of methods.
  - ML stays ML: translation only; not guided by SC.
  - SC stays SC: inference only; not guided by LLM.

Why it matters:

APPROACH

Combine machine learning with automated reasoning at the problem level.

- No augmentation of methods.
  - ML stays ML: translation only; not guided by SC.
  - SC stays SC: inference only; not guided by LLM.

Why it matters:

- Trustworthiness: the symbolic engine guarantees correctness; no hallucinations.

APPROACH

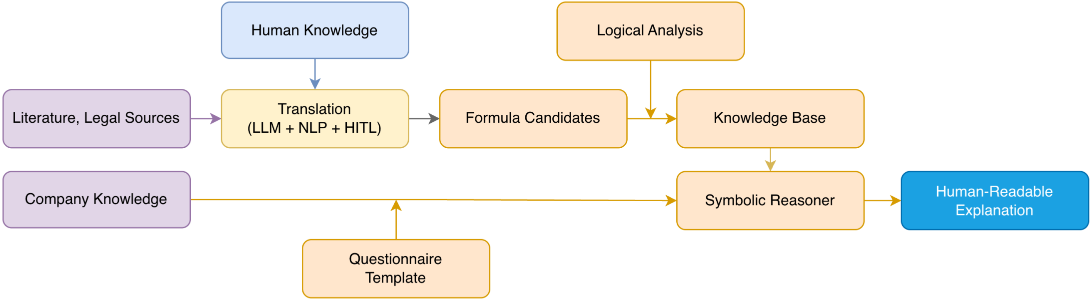
Combine machine learning with automated reasoning at the problem level.

- No augmentation of methods.
  - ML stays ML: translation only; not guided by SC.
  - SC stays SC: inference only; not guided by LLM.

Why it matters:

- Trustworthiness: the symbolic engine guarantees correctness; no hallucinations.
- Explainability: every proof step is explicit and can be inspected end to end.

ARCHITECTURE



LARGE LANGUAGE MODELS — THE CORE IDEA

What they are:

- Trained on vast text corpora to predict the next token.

LARGE LANGUAGE MODELS — THE CORE IDEA

What they are:

- Trained on vast text corpora to predict the next token.
- Fine-tuned on instruction/response pairs $\rightsquigarrow$ follow directives.

LARGE LANGUAGE MODELS — THE CORE IDEA

What they are:

- Trained on vast text corpora to predict the next token.
- Fine-tuned on instruction/response pairs $\rightsquigarrow$ follow directives.
- Further refined with human feedback $\rightsquigarrow$ reasoning, caution.

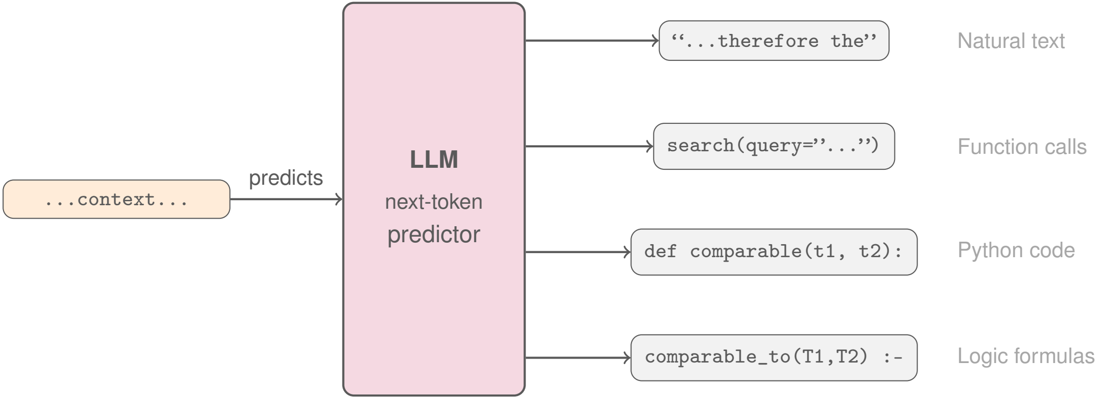
LARGE LANGUAGE MODELS — THE CORE IDEA

What they are:

- Trained on vast text corpora to predict the next token.
- Fine-tuned on instruction/response pairs $\rightsquigarrow$ follow directives.
- Further refined with human feedback $\rightsquigarrow$ reasoning, caution.

Key insight for us: A sufficiently capable next-token predictor can translate between languages — including from natural language into formal languages.

LLM — WHAT "NEXT TOKEN" CAN MEAN



STAGE 1: INFORMATION EXTRACTION

Structuring a legal norm before the LLM ever sees it.

Source — Glossary OECD TPG 2022

> "A controlled transaction is **comparable** to an uncontrolled transaction if none of the **differences between the transactions could materially affect** the price or margin examined, **or** if reasonably **accurate adjustments can be made** to eliminate the material effects of such differences."

STAGE 1: INFORMATION EXTRACTION

Structuring a legal norm before the LLM ever sees it.

Source — Glossary OECD TPG 2022

"A controlled transaction is **comparable** to an uncontrolled transaction if none of the **differences between the transactions could materially affect** the price or margin examined, **or** if reasonably **accurate adjustments can be made** to eliminate the material effects of such differences."

Extracted representation

ENTITIES	
Subject	controlled (T1)
Comparator	uncontrolled (T2)
C1	no material difference
C2	accurate adjustment possible
P	T1 comparable to T2
LOGICAL STRUCTURE	
Type	definition
Structure	$P \Leftrightarrow (C1 \vee C2)$

Goal: reduce ambiguity and make the logical structure explicit before translation.

STAGE 2: LLM TRANSLATION

From structured intermediate form to predicate logic.

Extracted representation

ENTITIES	
Subject	controlled (T1)
Comparator	uncontrolled (T2)
C1	no material difference
C2	accurate adjustment possible
P	T1 comparable to T2
LOGICAL STRUCTURE	
Type	definition
Structure	$P \Leftrightarrow (C1 \vee C2)$

STAGE 2: LLM TRANSLATION

From structured intermediate form to predicate logic.

Extracted representation

```
ENTITIES
Subject                 controlled (T1)
Comparator            uncontrolled (T2)
C1              no material difference
C2    accurate adjustment possible
P                 T1 comparable to T2

LOGICAL STRUCTURE
Type                        definition
Structure              P ⇔ (C1 ∨ C2)
```

Generated formula candidate

```
norm_source(comparable,
 ['OECDTPG2022Glossary'],
 [ en, 'A controlled transaction is comparable ...']).

equiv_def(
 comparable(T1, T2),
 and(
  controlled_transaction(T1),
  uncontrolled_transaction(T2),
  or(
   no_material_difference(T1, T2),
   accurate_adjustment_possible(T1)))).
```

STAGE 2: LLM TRANSLATION

From structured intermediate form to predicate logic.

Extracted representation

```
ENTITIES
Subject                controlled (T1)
Comparator           uncontrolled (T2)
C1               no material difference
C2     accurate adjustment possible
P                  T1 comparable to T2

LOGICAL STRUCTURE
Type                         definition
Structure              P ⇔ (C1 ∨ C2)
```

Generated formula candidate

```
norm_source(comparable,
 ['OECDTPG2022Glossary'],
 [ en, 'A controlled transaction is comparable ...']).

equiv_def(
 comparable(T1, T2),
 and(
  controlled_transaction(T1),
  uncontrolled_transaction(T2),
  or(
   no_material_difference(T1, T2),
   accurate_adjustment_possible(T1))))).
```

Guardrails restrict generation to syntactically valid formulas — no free-form output.

HUMAN-IN-THE-LOOP — REVIEW & SIGN-OFF

The formula candidate

```
norm_source(comparable,
 ['OECDTPG2022Glossary'],
 [ en, 'A controlled transaction
        is comparable ...']).

equiv_def(
 comparable(T1, T2),
 and(
  controlled_transaction(T1),
  uncontrolled_transaction(T2),
  or(
   no_material_difference(T1, T2),
   accurate_adjustment_possible(T1)))).
```

What the expert checks

- ✓ Predicate names match the shared ontology
- ✓ Appropriate use of connectives and quantifiers
- ✓ Sources cited recorded in `norm_source`
- ✓ Formula expresses accurately the meaning of the source text

HUMAN-IN-THE-LOOP — REVIEW & SIGN-OFF

The formula candidate

```
norm_source(comparable,
 ['OECDTPG2022Glossary'],
 [ en, 'A controlled transaction
        is comparable ...']).

equiv_def(
 comparable(T1, T2),
 and(
  controlled_transaction(T1),
  uncontrolled_transaction(T2),
  or(
   no_material_difference(T1, T2),
   accurate_adjustment_possible(T1))).
```

What the expert checks

- ✓ Predicate names match the shared ontology
- ✓ Appropriate use of connectives and quantifiers
- ✓ Sources cited recorded in `norm_source`
- ✓ Formula expresses accurately the meaning of the source text

No formula enters the reasoning engine without expert approval

THE REASONING COMPONENT — THEOREMA

Theorema offers:

- Human-readable (mathematical) input.

THE REASONING COMPONENT — THEOREMA

Theorema offers:

- Human-readable (mathematical) input.
- Predicate-logic reasoning with natural deduction-oriented inference calculus.

THE REASONING COMPONENT — THEOREMA

Theorema offers:

- Human-readable (mathematical) input.
- Predicate-logic reasoning with natural deduction-oriented inference calculus.
- Extensible inference calculus.

THE REASONING COMPONENT — THEOREMA

Theorema offers:

- Human-readable (mathematical) input.
- Predicate-logic reasoning with natural deduction-oriented inference calculus.
- Extensible inference calculus.
- Human-readable explainable output.

THEOREMA — EXAMPLE — LEGAL NORM

OECD TPG 2022

Glossary

DEFINITION (COMPARABLE TRANSACTIONS)

In[9]:= $\forall_{T1,T2,T}$

In[12]:=
```
comparable[T1, T2] :⟺
 controlledTransaction[T1] ∧ uncontrolledTransaction[T2] ∧
  (noMaterialDifference[T1, T2] ∨
   accurateAdjustmentPossible[T1, T2])
```
(*OECDTPG2022_Glossary_comparable*)

In[13]:=
```
controlledTransaction[T] :⟺ associated[from[T], to[T]]
```
(*OECDTPG2022_Glossary_controlledTransaction*)

In[14]:=
```
uncontrolledTransaction[T] :⟺ ¬ controlledTransaction[T]
```
(*OECDTPG2022_Glossary_uncontrolledTransaction*)

In[27]:=
```
noMaterialDifference[T1, T2] :⟺ ¬ (∃_D discriminate[D, T1, T2])
```
(*OECDTPG2022_Glossary_materialDifference*)

THEOREMA — EXAMPLE — COMPANY KNOWLEDGE

Case 01

Transaction

Facts (LoanT)

```
In[19]:= from[loan[T, S, 50, "A"]] := T        (lenderTS)
In[20]:= to[loan[T, S, 50, "A"]] := S          (borrowerTS)
```

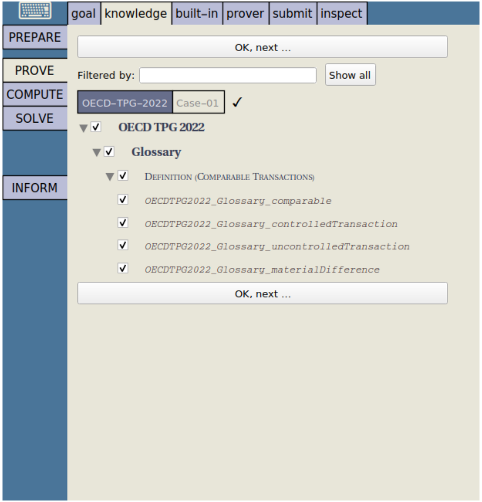
Facts (LoanI)

```
In[21]:= from[loan[I, S, 50, "A"]] := I        (lenderIS)
In[22]:= to[loan[I, S, 50, "A"]] := S          (borrowerIS)
```

Facts (LoanS)

In[28]:= $\forall_{x} \neg$ discriminate[x, loan[T, S, 50, "A"], loan[I, S, 50, "A"]]   (*nothing differs between loans*)

THEOREMA — EXAMPLE — KNOWLEDGE OVERVIEW



THEOREMA — EXAMPLE — PROOF OVERVIEW

Theorema Proof

▸ *Proof Simplification* ☆ Time spent for simplifying the proof: 0.005959s

We prove:

`comparable[loan[T, S, 50, "A"], loan[I, S, 50, "A"]]` (comparableLoan50)

under the assumptions:

$\forall_{T1,T2}$ `comparable`$[T1, T2]$:⟺ (OECDTPG2022_Glossary_comparable)

`controlledTransaction`$[T1]$ ∧ `uncontrolledTransaction`$[T2]$ ∧

(`noMaterialDifference`$[T1, T2]$ ∨ `accurateAdjustmentPossible`$[T1, T2]$),

We expand definitions:

In order to prove (comparableLoan50), using definition (OECDTPG2022_Glossary_comparable), we now show

`controlledTransaction[loan[T, S, 50, "A"]] ∧ uncontrolledTransaction[loan[I, S, 50, "A"]] ∧`
`(noMaterialDifference[loan[T, S, 50, "A"], loan[I, S, 50, "A"]] ∨`
`accurateAdjustmentPossible[loan[T, S, 50, "A"], loan[I, S, 50, "A"]]).` (G#60)

For proving (G#60) we prove the individual conjuncts.

☑ Proof of (G#60.1): ⊡

☑ Proof of (G#60.2): ⊡

☑ Proof of (G#60.3): ⊡

JKU

THEOREMA — EXAMPLE — PROOF BRANCH 1

Proof of (G#60.1):

We need to prove

```
controlledTransaction[loan[T, S, 50, "A"]].
```
(G#60.1)

We expand definitions:

In order to prove (G#60.1), using definition (OECDTPG2022_Glossary_controlledTransaction), we now show

```
associated[from[loan[T, S, 50, "A"]], to[loan[T, S, 50, "A"]]].
```
(G#62)

We apply substitutions:

In order to prove (G#62), using (lenderTS) and (borrowerTS), we now show

```
associated[T, S].
```
(G#63)

The goal (G#63) is identical to formula (T and S in same group) in the knowledge base. Thus, this part of the proof is finished.

THEOREMA — EXAMPLE — PROOF BRANCH 2

☑ Proof of (G#60.2):

We need to prove

```
uncontrolledTransaction[loan[I, S, 50, "A"]].
```
(G#60.2)

We expand definitions:

In order to prove (G#60.2), using definition (OECDTPG2022_Glossary_uncontrolledTransaction), we now show

```
¬ controlledTransaction[loan[I, S, 50, "A"]].
```
(G#64)

We prove (G#64) by contradiction, i.e. we assume

```
controlledTransaction[loan[I, S, 50, "A"]]
```
(A#65)

and derive a contradiction.

We expand definitions:

From (A#65) we know, by definition (OECDTPG2022_Glossary_controlledTransaction),

```
associated[from[loan[I, S, 50, "A"]], to[loan[I, S, 50, "A"]]]
```
(A#68)

We apply substitutions:

From (A#68) we know, by (lenderIS) and (borrowerIS),

```
associated[I, S]
```
(A#70)

This part of the proof is herewith finished, because (A#70) contradicts (I and S independent).

THEOREMA — EXAMPLE — PROOF BRANCH 3

☑ Proof of (G#60.3):

We need to prove

```
noMaterialDifference[loan[T, S, 50, "A"], loan[I, S, 50, "A"]] ∨
  accurateAdjustmentPossible[loan[T, S, 50, "A"], loan[I, S, 50, "A"]].
```
(G#60.3)

For proving the disjunction (G#60.3) we assume

```
¬ noMaterialDifference[loan[T, S, 50, "A"], loan[I, S, 50, "A"]]
```
(A_G#60.3.1)

and derive a contradiction.

We expand definitions:

From (A_G#60.3.1) we know, by definition (OECDTPG2022_Glossary_materialDifference),

$$\exists_{D}\ \mathtt{discriminate}[D, \mathtt{loan}[T, S, 50, \text{"A"}], \mathtt{loan}[I, S, 50, \text{"A"}]]$$
(A#74)

Since we know (A#74) we can take some D such that

```
discriminate[D, loan[T, S, 50, "A"], loan[I, S, 50, "A"]].
```
(A#76)

New terms have been detected.

We choose $x \to D$ in formula (nothing differs between loans) and obtain

```
¬ discriminate[D, loan[T, S, 50, "A"], loan[I, S, 50, "A"]].
```
(A#78)

This part of the proof is herewith finished, because (A#78) contradicts (A#76).

REASONING COMPONENT — EXPECTED CHALLENGES

- How to deal with contradicting legal norms.

REASONING COMPONENT — EXPECTED CHALLENGES

- How to deal with contradicting legal norms.
- How to cope with knowledge base size.

REASONING COMPONENT — EXPECTED CHALLENGES

- How to deal with contradicting legal norms.
- How to cope with knowledge base size.
- . . . ???

CONCLUSION

A hybrid architecture for explainable transfer pricing: LLMs translate legal norms into predicate logic; a symbolic engine reasons and explains every step.

- Problem-level coupling $\rightsquigarrow$ inference stays purely symbolic $\rightsquigarrow$ trustworthiness.

CONCLUSION

A hybrid architecture for explainable transfer pricing: LLMs translate legal norms into predicate logic; a symbolic engine reasons and explains every step.

- Problem-level coupling $\rightsquigarrow$ inference stays purely symbolic $\rightsquigarrow$ trustworthiness.
- Human-in-the-loop $\rightsquigarrow$ formula candidates must be approved.

CONCLUSION

A hybrid architecture for explainable transfer pricing: LLMs translate legal norms into predicate logic; a symbolic engine reasons and explains every step.

- Problem-level coupling $\rightsquigarrow$ inference stays purely symbolic $\rightsquigarrow$ trustworthiness.
- Human-in-the-loop $\rightsquigarrow$ formula candidates must be approved.
- Source-to-formula provenance $\rightsquigarrow$ explainability.

CONCLUSION

A hybrid architecture for explainable transfer pricing: LLMs translate legal norms into predicate logic; a symbolic engine reasons and explains every step.

- Problem-level coupling $\rightsquigarrow$ inference stays purely symbolic $\rightsquigarrow$ trustworthiness.
- Human-in-the-loop $\rightsquigarrow$ formula candidates must be approved.
- Source-to-formula provenance $\rightsquigarrow$ explainability.
- Local LLMs $\rightsquigarrow$ data sovereignty & scalability.

CONCLUSION

A hybrid architecture for explainable transfer pricing: LLMs translate legal norms into predicate logic; a symbolic engine reasons and explains every step.

- Problem-level coupling $\rightsquigarrow$ inference stays purely symbolic $\rightsquigarrow$ trustworthiness.
- Human-in-the-loop $\rightsquigarrow$ formula candidates must be approved.
- Source-to-formula provenance $\rightsquigarrow$ explainability.
- Local LLMs $\rightsquigarrow$ data sovereignty & scalability.

Status: ongoing project “TaxoLogic” (FFG “AI Ökosysteme 2025: AI for Tech & AI for Green”).

CONCLUSION

A hybrid architecture for explainable transfer pricing: LLMs translate legal norms into predicate logic; a symbolic engine reasons and explains every step.

- Problem-level coupling $\rightsquigarrow$ inference stays purely symbolic $\rightsquigarrow$ trustworthiness.
- Human-in-the-loop $\rightsquigarrow$ formula candidates must be approved.
- Source-to-formula provenance $\rightsquigarrow$ explainability.
- Local LLMs $\rightsquigarrow$ data sovereignty & scalability.

Status: ongoing project "TaxoLogic" (FFG "AI Ökosysteme 2025: AI for Tech & AI for Green").

Design and first prototype in progress.