Divide and Translate: Compositional First-Order Logic Translation and Verification for Complex Logical Reasoning


Hyun Ryu


Gyeongman Kim


Hyemin S. Lee


Eunho Yang

ryuhyun1905@kaist.ac.kr

Logical Reasoning

- What is a Logical reasoning?
  - A structured process of reaching conclusions from premises
  - One of the most challenging metrics to measure intelligence
- Large language models (LLMs) unlock the ability of machine to reason
- Chain-of-thought (CoT) prompting improves logical reasoning ability of LLMs
- However, CoT falls short in *complex* logical reasoning tasks which require long sequence of reasoning

Neurosymbolic Approaches

- Logic-LM, SatLM, LINC, etc. utilize an LLM with a symbolic solver
  1. LLM translates NL problem into satisfiability (SAT) problem that consists of FOL formulas
  2. Symbolic solver returns a mathematically correct solution
- By considering LLM as a semantic parser, it can avoid errors in reasoning steps
- Example

Each of five students—Hubert, Lori, Paul, Regina, and Sharon—will visit exactly one of three cities—Montreal, Toronto, or Vancouver—for the month of March, according to the following conditions: Sharon visits a different city than Paul. Hubert visits the same city as Regina. Lori visits Montreal or else Toronto. If Paul visits Vancouver, Hubert visits Vancouver with him. Each student visits one of the cities with at least one of the other four students.
Question: Which one of the following must be true?
(A) If any of the students visits Montral, Lori visits Montreal. (B) ...

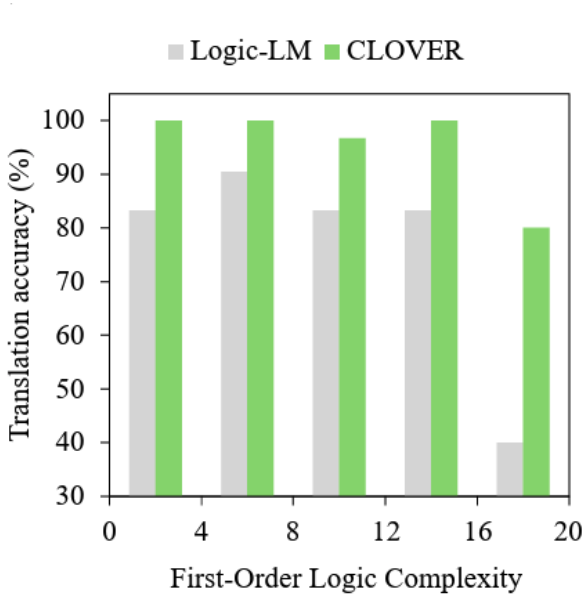
LLM as a semantic parser (NL-to-FOL)

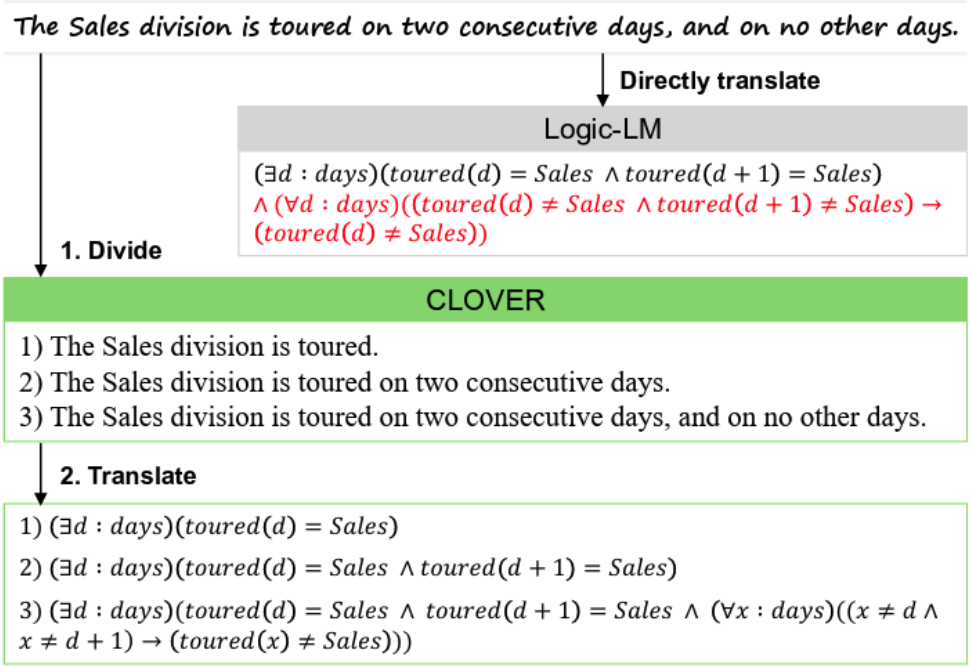
```
students = [Hubert, Lori, Paul, Regina, Sharon]
cities = [Montreal, Toronto, Vancouver]
visits = Function(students -> cities)

visits(Sharon) != visits(Paul)
Or(visits(Lori) == Montreal, visits(Lori) == Toronto)
ForAll([s1], Exists([s2], And(s2 != s1, visits(s1) == visits(s2))))
...
solve(Implies(Exists([s], visits(s) == Montreal), visits(Lori) == Montreal))
```

Limitations of Neurosymbolic Approaches

- LLM still cannot translate complex sentences
  - It falls short beyond a certain degree of FOL complexity
  - It correctly translates NL with simple logic, but fails to translate NL with more complex logic
- Our Idea: Divide and Translate!
  - First translate an atomic subsentence, and then translate more complex ones sequentially





Proposed Method: CLOVER

- But, how to divide the sentences?
- *Logical Dependency Parsing*
  - New parsing method for NL that represents FOL
  - Categorize sentence components into the following three:
    - Logic Units (U), Logic Couplers (C), and Logic Dependents (D)
  - Example
    - Target sentence: “Hamadi cannot be appointed to the same court as Perkins.”
    - Logical Dependency Structure:

Structure #1

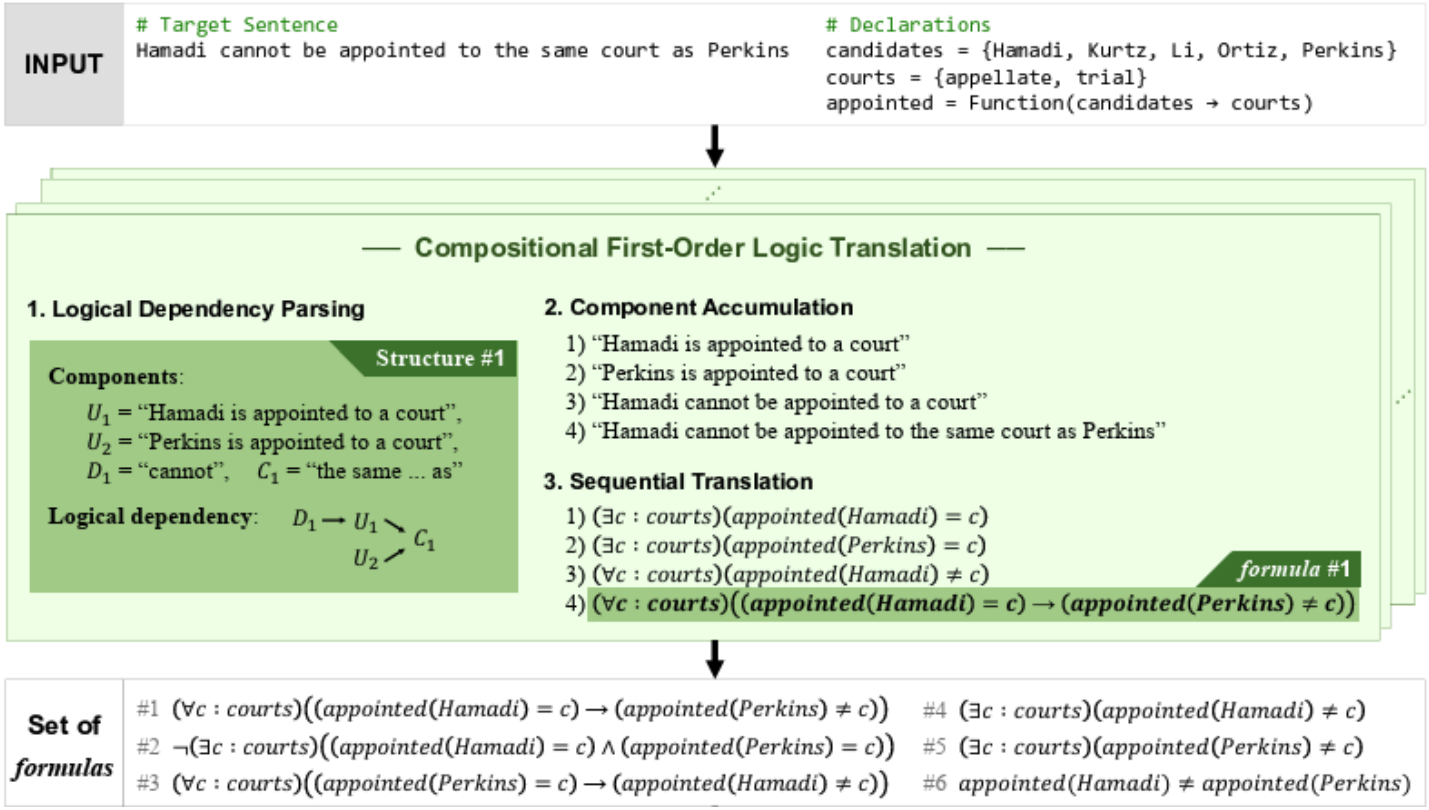
Components:

U_1 = “Hamadi is appointed to a court”,
U_2 = “Perkins is appointed to a court”,
D_1 = “cannot”, C_1 = “the same ... as”

Logical dependency: $D_1 \rightarrow U_1 \searrow C_1$, $U_2 \nearrow C_1$

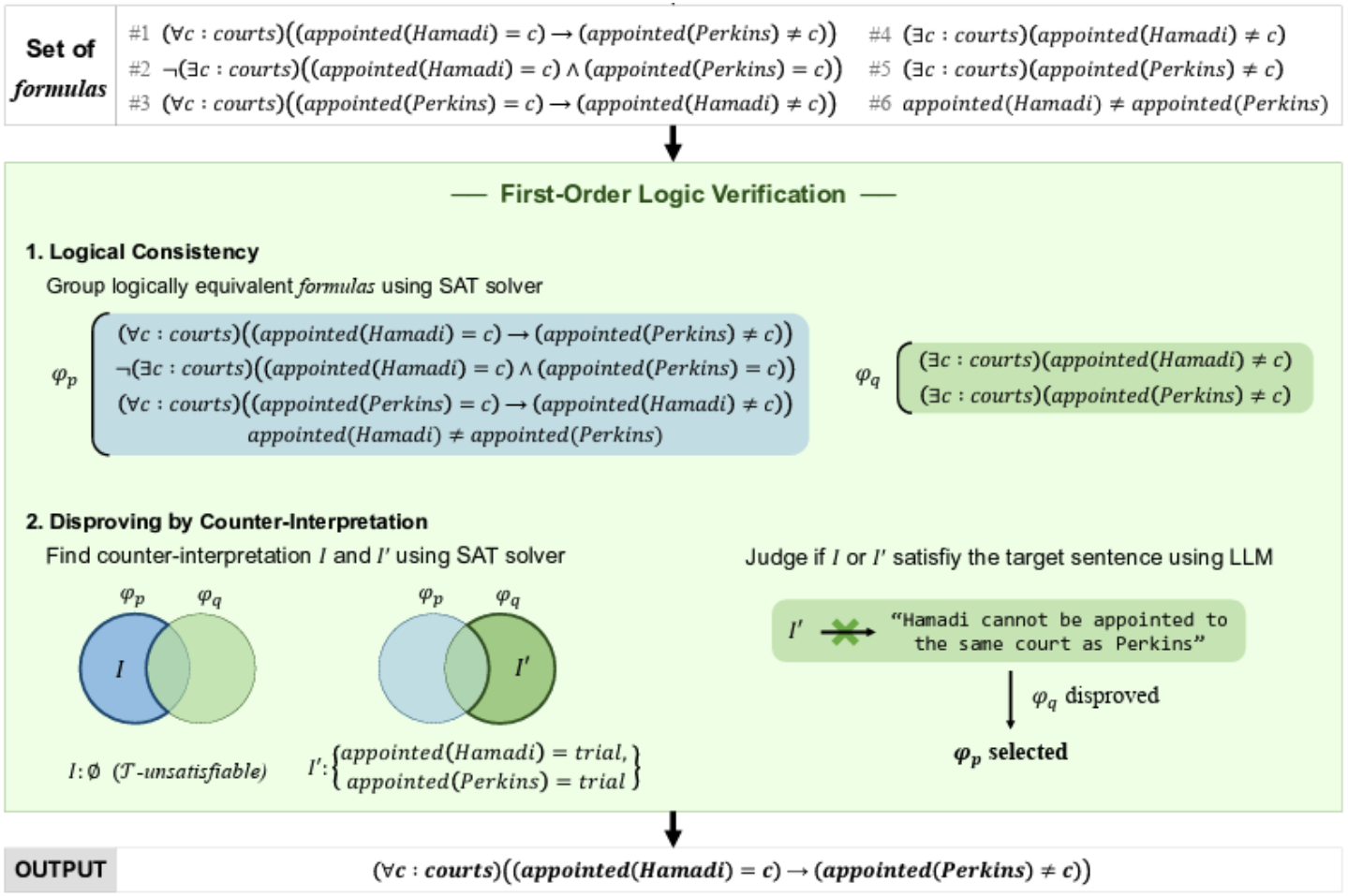
Proposed Method: CLOVER

- *Compositional First-Order Logic Translation*
  - Consists of three steps
  - There could be multiple output formulas, so we need to select the most reliable formula



Proposed Method: CLOVER

- *First-Order Logic Verification*
  - Two verification algorithms



Experiments

- Setup
  - Language models: gpt-4o, gpt-4o-mini, etc.
  - Baselines: Standard/CoT prompting and Neurosymbolic approaches
- CLOVER outperforms all the baselines!

	AR-LSAT	ZebraLogic	Puzzle	Symbol	Deduction	FOLIO	ProofWriter
Standard	30.3	0.4	63.0	74.7	84.7	70.9	53.7
CoT	36.8	0.4	51.0	80.8	94.0	73.9	78.0
SymbCoT	34.2	0.8	66.5	55.6	90.7	76.9	80.2
Logic-LM	42.4	45.4	64.0	81.8	95.3	75.4	95.3
CLOVER	**62.8**	**75.4**	**83.5**	**89.9**	**99.3**	**78.8**	**96.7**

	Program Acc		Execution Rate		Execution Acc	
	Logic-LM	CLOVER	Logic-LM	CLOVER	Logic-LM	CLOVER
AR-LSAT	17.3	**46.8**	33.8	**59.7**	51.3	**78.3**
Puzzle	60.0	**79.0**	79.5	**80.0**	75.5	**98.8**
Symbol	49.5	**76.8**	52.5	**82.8**	**94.2**	92.7
Deduction	92.7	**99.0**	97.3	**99.7**	95.2	**99.3**
FOLIO	51.2	**62.6**	65.5	**74.9**	78.2	**83.6**
ProofWrtier	94.2	**96.5**	96.8	**99.2**	97.2	**97.3**