Assessing the (In)Ability of LLMs to Reason in Interval Temporal Logic

Pietro Bellodi
Department of Mathematics and Computer Science, University of Ferrara, Italy

Pietro Casavecchia
Department of Mathematics and Computer Science, University of Ferrara, Italy

Alberto Paparella
Department of Mathematics and Computer Science, University of Ferrara, Italy

Guido Sciavicco
Department of Mathematics and Computer Science, University of Ferrara, Italy

Ionel Eduard Stan
Department of Informatics, Systems, and Communications, University of Milano-Bicocca, Italy

Abstract

The logical reasoning skills of Large Language Models (LLMs) is poorly understood and often overstated. Current evaluation suites rely on algebraic or commonsense puzzles that mix reasoning with symbolic manipulation and/or provide static datasets that quickly saturate or leak into pretraining corpora. In purely logical terms, the most relevant reasoning skill is the meta-mathematical task of valid formula recognition, which is at the foundation of higher-level reasoning tasks (including deduction and minimization of assertions, to name just a few). In the current landscape of LLMs benchmarking, puzzles are most often stated in propositional or first-order logic, with a few exceptions for point-based temporal logic, such as LTL; yet, in the real world, event-based temporal statements are prevalent, and they are more naturally expressed in interval-based temporal logic. Interval temporal logic offers a much richer (w.r.t. point-based temporal logic, for example) variety of problems, and not only do different languages present different expressive powers, but also the computational complexity of the validity problem can vary widely. In this paper, we tackle the problem of assessing the ability of LLMs to reason about interval-based statements in the form of validity recognition. We explore whether their accuracy is sensible to the underlying language, the computational complexity of the associated validity problem, and the intrinsic hardness of the problem in terms of formula length and modal depth of the problem. We benchmark several frontier LLMs (Gemma 3 27b It, Llama 4 Maverick, DeepSeek Chat V3 release 0324, Qwen 3 32b, and Qwen 3 235b) and show that, despite apparently impressive performance on algebraic or commonsense benchmarks, they falter on logically rigorous tasks.

2012 ACM Subject Classification Theory of computation → Modal and temporal logics; Theory of computation → Theory and algorithms for application domains

Keywords and phrases Large Language Models, Benchmarking, Interval Temporal Logic

Digital Object Identifier 10.4230/LIPIcs.TIME.2025.4

Supplementary Material *Software*: `https://github.com/aclai-lab/TIME2025-LLM`
archived at `swh:1:dir:de0d3840df7e24429dd48c845a7e784aa32e4da2`

Acknowledgements We acknowledge the support of the FIRD project Methodological Developments in Modal Symbolic Geometric Learning, funded by the University of Ferrara.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success across a wide range of natural language tasks in recent years. Models like GPT-3 [7] and its successors demonstrated emergent capabilities in reasoning and problem-solving when prompted appropriately [33]. Notably, benchmarks such as GSM8K [10] and MATH [16] spurred progress in arithmetic

and mathematical reasoning by encouraging multi-step chain-of-thought solutions. However, the question of whether LLMs can *reliably perform logical reasoning*, in the rigorous sense of formal logic, remains underexplored and challenging [27, 25]. Logical consistency and deductive inference are critical for advanced AI reasoning, yet even state-of-the-art models often stumble on tasks requiring strict logical entailment, especially in the presence of negation or complex rules [25, 21].

There is growing evidence that current LLMs struggle with basic logical deductive questions that humans find trivial. For example, a recent evaluation called SimpleBench showed that non-expert humans significantly outperform frontier LLMs on a set of 200 straightforward reasoning questions involving spatial-temporal reasoning and logical trick questions [2]. Similarly, general intelligence tests like the Abstraction and Reasoning Corpus (ARC) [8], which requires solving abstract visual puzzles independent of prior knowledge, remain far from solved by machines, underscoring the gaps in core reasoning abilities. These observations highlight the need for systematic evaluation of LLMs' logical reasoning skills, beyond the realms of arithmetic or commonsense reasoning.

In this work we consider the problem of benchmarking the ability of LLMs to reason about temporal events and their relationships. Our approach automatically generates instances from first principles, using logic tautologies and formal inference rules, so that each example's label (valid or invalid formula) is guaranteed correct by construction. By leveraging the well-defined semantics of formal logics, we avoid the ambiguities and potential errors of human-crafted logical puzzles, providing a *reliable ground truth* for model evaluation. In our specific testbed we focus on Halpern-Shoham interval temporal logic (HS) [14], which can be considered a standard logical setting for event-based reasoning at the qualitative level. Reasoning in HS can be considered a hard problem; depending on the specific subset of operators that occur in a formula, establishing its validity status is a problem whose complexity ranges from NP-complete, to PSPACE-complete, to EXPSPACE-complete, to non primitive recursive (NPR)-complete, to undecidable. In the particular case of linear models based on the set of natural numbers, the status of every possible syntactic fragment is reported in [5].

Our approach fundamentally *differs from existing work* in three ways. First, we target *logical validity in a formal sense*, rather than numerical or symbolic manipulation. Many prior reasoning benchmarks for LLMs (e.g., math word problems in GSM8K/MATH, or code execution tasks) involve algebraic reasoning or pattern matching that, while complex, do not probe a model's ability to apply abstract logical rules or handle operators like negation and implication in a principled way. In contrast, our evaluation specifically stresses logical consistency and the handling of negation, which has been identified as a stumbling block for LLMs' reasoning [25]. Second, we focus on event-based temporal reasoning, so far neglected in the context of LLMs benchmarking. Third, we explore LLMs reasoning ability in relation to problem length (i.e., number of involved symbols), problem abstraction level (i.e., modal depth), and intrinsic problem complexity (i.e., computational class to which it belongs). Fourth, we employ an algorithmic test generator that produces valid formulas, rather than a fixed set of predetermined ones, avoiding the risk of future leaking into training datasets. We validate our approach by conducting an extensive evaluation of several leading LLMs on the generated logical reasoning tasks. In particular, we benchmark a suite of state-of-the-art models, namely Gemma 3 27b It, Llama 4 Maverick, DeepSeek Chat V3 release 0324, Qwen 3 32b, and Qwen 3 235b. Through systematic experiments on these diverse systems, we analyze their performance on problems that require genuine logical reasoning. As we show later, even the best models *struggle on logically challenging instances*, confirming findings from recent studies that current LLMs have not attained robust logical competence [21, 25, 30].

In a sense, a test of reasoning capabilities as we designed it is a test of *general intelligence* level. We can safely assume that an LLM *does not know* what interval-based temporal logic is in the strict sense (the amount of existing and available material on this topic is exponentially lower than, for example, linear time point-based temporal logic); for the test purposes such a notion is therefore *explained* to the LLM (via prompting); finally the ability of the LLM to reason about what was explained are *tested*. A positive result from a test so designed would not be a proof that a model possesses general intelligence; a negative one would be a proof that the model does not.

2 Related Work

The quest to measure and improve reasoning in LLMs has led to a variety of benchmarks targeting different reasoning facets.

On the numerical side, the Grade School Math 8K dataset (GSM8K) [10] and the MATH competition dataset [16] have become standard tests for arithmetic and mathematical problem solving. These datasets require multi-step reasoning and have spurred innovations like chain-of-thought prompting [17, 33] to elicit latent reasoning steps from models. More generally, BIG-bench and related efforts compiled diverse reasoning tasks, like commonsense and symbolic, to probe emerging capabilities of large models [28]. However, *formal logical reasoning* was not a primary focus in these early benchmarks. Notably, ARC challenge [8] targeted abstract pattern reasoning via visual puzzles to evaluate general intelligence; LLMs have struggled with ARC-style tasks unless augmented with specialized tools, reflecting a gap in out-of-distribution reasoning. Recently, the SimpleBench evaluation explicitly highlighted fundamental reasoning gaps in frontier models: on a suite of basic logic, spatial, and trick questions, human participants achieved over 80% accuracy while even top-tier LLMs (e.g., o3, Claude Sonnet, Gemini, Grok 3, and DeepSeek R1) remained far lower (30–50% range), often failing on problems requiring careful logical consistency. These findings motivate the development of dedicated logical reasoning benchmarks.

A different line of work has emerged to directly evaluate (and train) models on logical deduction tasks using controlled datasets. A pioneering example is RuleTaker [9], which generated synthetic natural language facts and rules and quizzed models on deductive conclusions. Remarkably, transformers fine-tuned on RuleTaker showed the ability to correctly answer many queries, even generalizing to some deeper inference chains, hinting at the possibility of learned logical reasoning. Subsequent efforts extended this approach: LogicNLI [31] expanded the scope to first-order logic, and ProofWriter [29] augmented the RuleTaker paradigm by asking models not only for yes/no answers but also to generate explicit natural language proofs for their conclusions. ProofWriter demonstrated that models could produce plausible step-by-step derivations for synthetic logic puzzles, though evaluation of proof correctness remained difficult. Saparov and He proposed PrOntoQA [27], another synthetic QA dataset that encodes formal deductive reasoning problems, used it to formally analyze how models reason, and founding that LLMs tend to be superficial reasoners often jumping to conclusions that are logically invalid if they appear superficially plausible. In addition to fully synthetic data, there have been expert-crafted datasets to test logical reasoning. A notable benchmark is FOLIO [15], introduced by Han et al., which consists of logically complex natural language puzzles written by experts and annotated with first-order logic forms. FOLIO problems are open-domain and diverse, covering, among others, quantifiers and implications, intended to require genuine logical deduction from the given premises. Models like GPT-3 and PaLM were reported to perform poorly on FOLIO, indicating that pretraining alone does not equip LLMs

with robust logic skills [15]. Another example is the AR-LSAT corpus [35], which contains analytical reasoning questions from law school admission tests; these are high-level logical puzzles in natural language, and zero-shot GPT-4 still finds many of them challenging [34]. One last contribution is [18], that proposes an automatic test generator (ATG), similar to our proposal, but limited to propositional logic. Overall, these works illustrate a spectrum from purely synthetic logic exercises to realistic logical reasoning problems. Across the board, *negation and multi-step inference* emerge as common pain points for current models [15, 25].

More recently, researchers have started assembling more systematic benchmarks to thoroughly probe LLMs' logical reasoning across multiple phenomena. Parmar et al. introduce LogicBench [25], a collection of 25 distinct logical reasoning patterns expressed in natural language. Each LogicBench question focuses on a single inference rule (e.g. modus ponens, modus tollens, syllogism, transitivity, etc.) either in propositional, first-order, or non-monotonic logic, presented as a small textual scenario with a yes/no question. This controlled setup allows one to pinpoint which specific forms of inference a model handles or fails. Testing GPT-4, ChatGPT, Gemini, Llama-2 and others, they found that existing LLMs do not fare well on LogicBench – especially on instances involving more complex reasoning or embedded negations, performance was near chance. Our work is closely aligned with this goal of systematic evaluation, though we approach it by generating formula-based entailment instances from formal semantics (as opposed to natural language templated questions). Another recent benchmark, LTLBench [30], specifically targets temporal logic reasoning. Tang and Belle developed a pipeline using random graph generation and an LTL model checker to create 2,000 temporal reasoning challenges, and evaluated six LLMs on them. Their results showed that while some LLMs exhibit basic competence on simple temporal queries, they struggle as the complexity increases (e.g. more events or nested temporal operators) and substantially underperform compared to what would be required for sound temporal reasoning. LTLBench demonstrates the feasibility of combining formal verification tools with LLM evaluation – an approach our work generalizes and extends to other logics. In a similar vein, Morishita et al. present the Formal Logic Deduction benchmark (FLD) [21], generated from a complete set of first-order logic deduction rules. They report that even GPT-4 solves only about half of the problems in FLD, underlining that pure logical deduction (even when posed in natural language) remains a serious challenge for LLMs.

Beyond temporal logic, reasoning about space and structured knowledge are important dimensions of logical evaluation. Spatial relation formalisms such as RCC5 and RCC8 [26] provide a calculus for qualitative spatial reasoning (e.g. relations like disjoint, overlap, and containment between regions). These have not yet been widely used to evaluate LLMs, but they present an attractive next step: one could generate spatial scenarios and queries in natural language underpinned by RCC constraints, and test if LLMs can infer implied spatial relations. We posit that our methodology can be applied here by generating spatial logic formulas with known entailments. Similarly, Description Logics (DL) underpin knowledge graphs and ontologies in the Semantic Web, enabling rigorous inference of subclass relations, instance membership, etc. [4]. Traditional AI systems employ DL reasoners (like FaCT++ or HermiT) to perform these inferences reliably. In contrast, an LLM might be used to answer ontology queries or complete a knowledge graph, but concerns arise about whether it can honor the formal logical constraints (e.g. avoid asserting mutually inconsistent facts). There is ongoing research in combining LLMs with symbolic reasoners to ensure logical consistency in knowledge-intensive applications. These neuro-symbolic approaches typically involve translating natural language to a logical form, using a logic reasoner to derive conclusions or check consistency, and then translating back to text [24, 23].

A consistent observation across these benchmarks is that negation and non-monotonic reasoning are weak spots for LLMs. For instance, LogicBench finds that models often misunderstand statements with negated conditions or conclusions [25]; similarly, PrOntoQA analysis noted that models are apt to assume a fact is true unless explicitly contradicted, even if logically it should be undetermined [27]. This tendency relates to the shallow heuristics LLMs might pick up from text, which break down for logical constructs like negation that require careful semantic interpretation. The broader implication is that purely neural models alone may lack the guarantees of logical soundness that symbolic reasoning provides. By developing benchmarks grounded in formal logic (as we do in this paper), we contribute toward bridging this gap. A robust evaluation methodology for LLMs logical reasoning is not only academically interesting but also practically vital as these models begin to be deployed in areas like legal reasoning, safety-critical decision making, and knowledge graph completion, where logical correctness is paramount. Our work specifically addresses this by including a variety of entailment cases with negated formulas and ensuring that only logically valid inferences count as correct. We thereby force models to confront the full truth-functional meaning of negation and other operators. Another aspect is combinatorial complexity: multi-step logical reasoning (combining several premises) taxes the models' limited reasoning depth and working memory. Datasets like ProofWriter and FLD explicitly vary the number of inference steps, and performance drops as steps increase [29, 21]. In our evaluation, we similarly consider entailments that may require reasoning across multiple temporal steps or combining multiple logical conditions. This allows us to examine whether models can perform reasoning beyond one-hop inference in a formal setting.

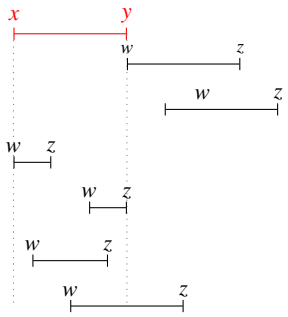
3 Interval Temporal Logic

While several different interval temporal logics have been proposed in the recent literature [13], *Halpern and Shoham's Modal Logic for Time Intervals (HS)* [14] is certainly the formalism that has received the most attention. Let $\mathbb{D} = \langle D, < \rangle$ be a linear order with *domain* D; in the following, we shall use D and $\mathbb{D}$ interchangeably. A *strict interval* over $\mathbb{D}$ is an ordered pair $[x, y]$, where $x, y \in \mathbb{D}$ and $x < y$. If we exclude the identity relation, there are 12 different binary ordering relations between two strict intervals on a linear order, often called *Allen's interval relations* [3]: the six relations R_A (*adjacent to*, also known as *after*), R_L (*later than*), R_B (*begins*, also known as *starts*), R_E (*ends*, also known as *finishes*), R_D (*during*) and R_O (*overlaps*), depicted in Tab. 1, and their *inverses*, that is, $R_{\overline{X}} = (R_X)^{-1}$, for each $X \in \{A, L, B, E, D, O\}$. We interpret interval structures as Kripke structures, with Allen's relations playing the role of accessibility relations. Thus, we associate an *existential modality* $\langle X \rangle$ with each Allen's relation R_X. Moreover, for each $X \in \{A, L, B, E, D, O\}$, the *transpose* of modality $\langle X \rangle$ is the modality $\langle \overline{X} \rangle$ corresponding to the inverse relation $R_{\overline{X}}$ of R_X. Now, let $\mathcal{X} = \{A, \overline{A}, L, \overline{L}, B, \overline{B}, E, \overline{E}, D, \overline{D}, O, \overline{O}\}$; well-formed HS formulas are built from a set of *propositional letters* $\mathcal{P}$, the classical connectives $\vee$ and $\neg$, and a modality for each Allen's interval relation, as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle X \rangle \varphi,$$

where $p \in \mathcal{P}$ and $X \in \mathcal{X}$. The other propositional connectives and constants (i.e., $\psi_1 \wedge \psi_2 \equiv \neg(\neg\psi_1 \vee \neg\psi_2), \psi_1 \rightarrow \psi_2 \equiv \neg\psi_1 \vee \psi_2$ and $\top = p \vee \neg p$), as well as, for each $X \in \mathcal{X}$, the *universal modality* $[X]$ (e.g., $[A]\varphi \equiv \neg\langle A \rangle \neg\varphi$), can be derived in the standard way. The set of all subformulas of a given HS formula φ is denoted by $sub(\varphi)$.

Table 1 Allen's interval relations and HS modalities.

HS modality	Definition w.r.t. the interval structure	Example
$\langle A \rangle$ (adjacent)	$[x,y]R_A[w,z] \Leftrightarrow y = w$	
$\langle L \rangle$ (later)	$[x,y]R_L[w,z] \Leftrightarrow y < w$	
$\langle B \rangle$ (begins)	$[x,y]R_B[w,z] \Leftrightarrow x = w \wedge z < y$	
$\langle E \rangle$ (ends)	$[x,y]R_E[w,z] \Leftrightarrow y = z \wedge x < w$	
$\langle D \rangle$ (during)	$[x,y]R_D[w,z] \Leftrightarrow x < w \wedge z < y$	
$\langle O \rangle$ (overlaps)	$[x,y]R_O[w,z] \Leftrightarrow x < w < y < z$	

The strict semantics of HS is given in terms of *interval models* of the type $M = \langle \mathbb{I}(\mathbb{D}), V \rangle$, where $\mathbb{D}$ is a linear order, $\mathbb{I}(\mathbb{D})$ is the set of all strict intervals over $\mathbb{D}$, and V is a *valuation function* $V : \mathcal{P} \to 2^{\mathbb{I}(\mathbb{D})}$ which assigns to every atomic proposition $p \in \mathcal{P}$ the set of intervals $V(p)$ on which p holds. The truth of a formula φ on a given interval $[x,y]$ in an interval model M, denoted by $M, [x,y] \Vdash \varphi$, is defined by structural induction on the complexity of formulas, as follows:

$$
\begin{array}{lll}
M, [x,y] \Vdash p & \text{if and only if} & [x,y] \in V(p), \text{ for each } p \in \mathcal{P}, \\
M, [x,y] \Vdash \neg\psi & \text{if and only if} & M, [x,y] \nVdash \psi, \\
M, [x,y] \Vdash \psi_1 \vee \psi_2 & \text{if and only if} & M, [x,y] \Vdash \psi_1 \text{ or } M, [x,y] \Vdash \psi_2, \\
M, [x,y] \Vdash \langle X \rangle \psi & \text{if and only if} & \text{there exists } [w,z] \text{ s.t. } [x,y]R_X[w,z] \text{ and } M, [w,z] \Vdash \psi,
\end{array}
$$

where $X \in \mathcal{X}$. Given a model $M = \langle \mathbb{I}(\mathbb{D}), V \rangle$ and a formula φ, we say that M *satisfies* φ if there exists an interval $[x,y] \in \mathbb{I}(\mathbb{D})$ such that $M, [x,y] \Vdash \varphi$. A formula φ is *satisfiable* if there exists an interval model that satisfies it. Moreover, a formula φ is *valid* if it is satisfiable at every interval of every (interval) model or, equivalently, if its negation $\neg\varphi$ is *unsatisfiable*.

By setting $\mathbb{D} = \mathbb{N}$, we limit our attention to interval models based on the set of natural numbers. This is not the only scenario that has been studied in the context of HS, but it is a very common one; it is the interval counterpart to the typical interpretation of LTL on the same domain. The satisfiability problem for HS is undecidable, and a great amount of effort has been devoted to the search of well-behaved syntactic fragments of it. The result of such an effort, in the case of natural numbers, is summarized in [5], and pictured in Fig. 1.

Interval temporal logic is an important tool in formal reasoning about temporal events. It is applied in several areas of artificial intelligence and machine learning (see, e.g., [19, 20], and being able to correctly reason in such a language can be of relevance. In the past, sound and complete tableau systems have been introduced in prototypical form in [6, 11, 12, 22] for variants, fragments, and generalizations of HS; however, the problem of reasoning in HS is still open in practical terms. While reasoning tasks can vary, it is known that most of them can be reduced to validity recognition, which is therefore representative of the reasoning challenges that a specific logical system poses. The question we pose is whether LLMs are able to establish if a given HS-formula is valid, and if their accuracy is sensible to the intrinsic difficulty of the problem. Such difficulty can be measured in several ways, including the length of the formula, its modal depth, and the computational complexity class to which the smallest fragment that contains the formula belongs to.

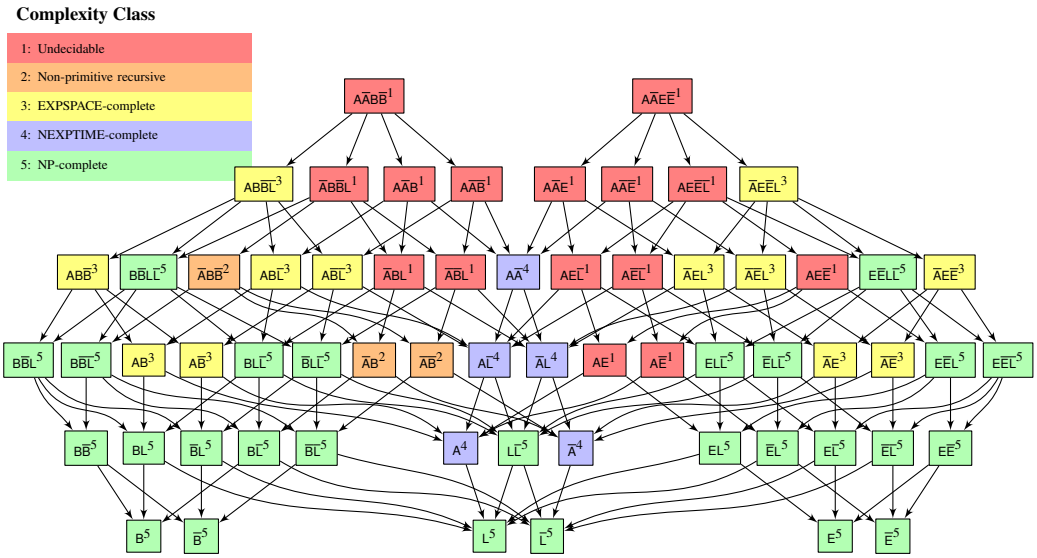


Figure 1 Relative expressive power and computational complexity of fragments of HS interpreted on models based on $\mathbb{N}$; unreported fragments are undecidable.

4 Benchmark Generation

The key point of our problem generation approach is the observation that *reasoning corresponds to validity recognition*. By their own nature, LLMs convey the idea of *natural language* reasoning, that is, the idea of reaching some logical conclusion from some set of premises. In turn, this reflects the concept of *logical* reasoning. However, while in most cases existing approaches to LLMs benchmarking relay on *common sense* logic, an automatic and systematic approach suggests the uses of *formal* logic. As a consequence, one should be easily convinced that testing reasoning capabilities corresponds to testing the ability of a system to identify a valid assertion, which is, by definition, a valid formula. The nature of LLMs to seemingly comprehend natural language should therefore not be seen as a limit, i.e., by focusing on testing common sense, natural language reasoning instances, but as an opportunity to explore their ability of following instructions, such as, given a sound and complete explanation of a chosen formal logic system, identify whether a certain reasoning is valid in it, that is, identify whether a given formula is valid. Moreover, the practice of testing and using LLMs to deal with code, such as LaTex code, programming code, Markdown, and tasking models with writing, correcting, completing, and modifying it, is now folklore. In the same spirit, the idea of testing LLMs with formal logic should be considered natural, and it should not be criticized as *unnatural*. The question we pose is: can formal reasoning tasks be carried out with distributional semantics?

Automatic theorem generation is a simply defined problem: given a set of theorems, produce a new theorem. However, it is also an ill-defined one, as it is unclear what constitutes an *interesting* theorem, especially from the point of view of its proof. While there exist attempts at solving this problem in propositional logic [18], automatic theorem generation is at its initial research stage (unlike, for example, automatic theorem proving) and, as it seems, there are no available systems for the case of modal, and in particular temporal case.

The starting point for theorem generation is existing theorems or axioms. In classic axiomatic theory, new theorems are generated by applying *sound deduction rules* to existing ones; classical deduction rules include *modus ponens*, *universal generalization*, and *uniform*

substitution. For most modal, temporal, and spatial logics that do have a sound and complete finite axiomatization, the latter is based on the above rules only; in some cases, such as that of HS, other, non-standard, rules must be added. It is well-known that Hilbert-style deduction system does not excel in producing very intuitive proofs, unlike other systems such as natural deduction (however, while Hilbert-style axiomatic systems have been studied for several logics, there exist essentially no natural deduction systems other than for propositional and first-order logics, plus some few minor exceptions). The purpose of an axiomatic system is to be able to produce a proof of a valid formula, and, as a consequence, to prove that in fact a given formula is valid, while the purpose of an automatic theorem generator is that of producing new valid formulas from existing ones, and an immediate algorithm reduces the latter to the former. In the case of HS, known validities in the language of HS come from three sources, that is, the original axiomatic system for HS [32], the axiomatic system for the fragment A$\overline{\text{A}}$ [12], and the collection of inter-definability of operators presented in [1] (examples of axioms can be seen in Tab. 2), and the process can be described as follows. Let $\mathcal{L}$ be a collection of valid HS-formulas, and $\mathcal{S}$ a collection of random well-formed HS-formulas, and apply one of the following rules:

- **i)** *uniform substitution*: choose a random validity $\varphi \in \mathcal{L}$, a random formula $\psi \in \mathcal{S}$, and a propositional letter p that occurs in φ, and produce the formula $\varphi[p/\psi]$;
- **ii)** *universal generalization*: choose a random validity $\varphi \in \mathcal{L}$ and a universal modality $[X]$, and produce the formula $[X]\varphi$;
- **iii)** *modus ponens*: choose two random validities $\varphi, \varphi \to \psi \in \mathcal{L}$, and produce the formula ψ.

▶ **Proposition 1.** *Given a set of valid HS-formulas $\mathcal{L} = \{\varphi_1, \ldots, \varphi_n\}$ and a set of well-formed HS-formulas $\mathcal{S}$, one application of the above algorithm produces a valid formula φ_{n+1}.*

The above algorithm produces valid formulas of the type $\varphi \to \psi$, with arbitrary syntactical complexity. Well-knowingly, (modal, temporal) logical formulas with Boolean semantics can be valid, if they are satisfied in every model (and world), contradictory, if they are never satisfied, or contingencies, if they are not valid nor contradictory. In this work, we focused on the ability of a LLM to distinguish between valid and contradictory formulas. In order to generate a random contradictory formula, it suffices to negate a valid one generated by the above algorithm; however, this creates a clear syntactic difference between the two classes, which may create bias towards one of the two classes. To circumvent this problem, we applied the following strategy:

- **i)** we produced a set S of valid formulas of the type $\varphi \to \psi$, randomly partitioned into two sets S_v and S_c;
- **ii)** we replaced every formula $\varphi \to \psi$ in S_v by its equivalent one $\neg\neg(\varphi \to \psi)$;
- **iii)** we replaced every formula $\varphi \to \psi$ in S_c by its opposite one $\neg(\varphi \to \psi)$;
- **iv)** finally, for every resulting formula in both S_v and S_c, we applied standard transformation rules to progressively push the negation symbols within the formula, up to a randomly chosen level.

As a result, formulas in both S_v and S_c have a non-predefined syntactical aspect, eliminating the risk of syntactic bias.

5 Results and Discussion

We approached this problem using the standard prompting techniques *context* (*ctx*), *few shots* (*fs*) [7], and *chain of thought* (*cot*) [33], combining them in a systematic way. As a form of baseline, we also prompted each model with no instructions, except the question itself; we

Table 2 Examples of axioms used to generate HS-theorems.

Axiom	Comment
all propositional validities	
$\langle A\rangle\langle A\rangle p \rightarrow \langle L\rangle p$	*definition of later*
$\langle B\rangle\langle E\rangle p \leftrightarrow \langle D\rangle p$	*definition of during*
$\langle \overline{B}\rangle\langle \overline{E}\rangle p \leftrightarrow \langle \overline{D}\rangle p$	*definition of during*
$\langle B\rangle\langle B\rangle p \leftrightarrow \langle B\rangle p$	*transitivity of starts*
$\langle A\rangle\langle A\rangle\langle A\rangle p \leftrightarrow \langle A\rangle\langle A\rangle p$	*pseudo-transitivity of meets*
$\langle B\rangle\langle E\rangle p \leftrightarrow \langle E\rangle\langle B\rangle p$	*commutativity of starts/finishes*

refer to this technique as *barebone*; on the other hand, *chain of thought*, *few shots*, and *context* are combined in the 8 possible ways, whereas the minimal configuration corresponding to a context without instructions is referred to as *base*, obtaining, in the end, 9 different prompts per single problem.

Taken individually, the prompts we used are as follows. The *barebone* baseline:

Given an interval temporal logic formula in the language of Halpern and Shoham's Modal Logic of Allen's Relations, reply with uppercase "[VALID]" if the formula is valid or uppercase "[INVALID]" if it is not.

Then, we designed the following *context*, structured, in turn, into the sections *purpose, syntax, semantics, task,* and *objective*:

*## **Purpose**
HS is a formal system for reasoning about interval-based events on a linear model based on the natural numbers. This context will define HS's syntax and semantics. The ultimate goal is to check if a HS formula is logically valid.
Syntax of HS
Propositional Letters
Let AP be a countable set of atomic propositions (p, q, r, ...), representing basic facts.
Well-Formed Formulas (wffs)
HS formulas are built inductively:*

*- **Base case**: Every p in AP is a wff.
- **Inductive cases**: If φ and ψ are wffs, then so are:*
...
*## **Semantics over Infinite Traces** Formulas of HS are interpreted over interval models based on the natural numbers N. Define I(N) as the set of all intervals [x,y] where x and y are natural numbers and x<y, and V as a function that assigns to each interval [x,y], the subset of AP of all and only propositional letters that are true on [x,y]. A model M is a pair (I(N),V). The satisfaction relation ** M,[x,y]|= φ ** for a model M and an interval [x,y] is defined by induction on the formula:*

*- **Atomic Propositions:**
- M,[x,y] |= p if and only if p belongs to V([x,y]), for all atomic propositions p in AP.
- **Boolean Operators:***
...
*## **Task: Evaluate HS Formula Validity***

Table 3 Overall accuracy in positive (TP) and negative (TN) cases, and overall average accuracy (AC) per model and prompt configuration.

	Gemma 3 27b It			Llama 4 Maverick			DeepSeek Chat V3			Qwen 3 32b			Qwen 3 235b		
	TP	TN	AC	TP	TN	AC	TP	TN	AC	TP	TN	AC	TP	TN	AC
barebone	0.17	0.92	0.55	0.69	0.72	0.71	0.39	0.92	0.65	0.41	0.93	0.67	0.50	0.72	0.61
base	0.07	0.97	0.52	0.35	0.91	0.62	0.04	1.00	0.52	0.12	0.96	0.54	0.51	0.73	0.62
ctx	0.09	0.96	0.52	0.47	0.78	0.62	0.05	1.00	0.53	0.09	0.98	0.54	0.34	0.88	0.61
cot	0.20	0.97	0.58	0.47	0.86	0.67	0.55	0.91	0.73	0.41	0.96	0.69	0.45	0.94	0.69
fs	0.48	0.80	0.64	0.83	0.73	0.77	0.53	0.86	0.69	0.71	0.55	0.63	0.80	0.60	0.70
ctx+cot	0.19	0.95	0.57	0.54	0.83	0.69	0.54	0.92	0.73	0.43	0.97	0.70	0.48	0.94	0.71
ctx+fs	0.48	0.68	0.58	0.50	0.79	0.65	0.58	0.86	0.72	0.45	0.75	0.60	0.75	0.66	0.70
cot+fs	0.26	0.94	0.60	0.87	0.78	0.82	0.58	0.84	0.71	0.48	0.93	0.71	0.49	0.88	0.68
ctx+cot+fs	0.30	0.92	0.61	0.83	0.78	0.81	0.64	0.85	0.75	0.47	0.93	0.71	0.61	0.84	0.72
average	0.25	0.90	0.57	0.62	0.80	0.71	0.43	0.91	0.67	0.40	0.88	0.64	0.55	0.80	0.67

*### **Objective**
Determine whether the formula is valid using HS semantics and reasoning. The formula can be written using symbols for atomic propositions (e.g., p, q, r, ...), negation operator (i.e., !), conjunction operator (i.e., &), ...*

The *objective* section when we prompted the models without chain of thought has the following structure:

*### **Instructions**
Reply **only** with uppercase "[VALID]" if the formula is valid or uppercase "[INVALID]" if it is not. **Do not explain your reasoning**.*

When using *chain of thought* the latter becomes:

*### **Instructions**
Follow these steps rigorously:
1. **Parse the Formulas**: Identify operators and subformulas.
2. **Apply Semantics**: Check if the formula necessarily holds in all infinite traces.
3. **Construct Proof/Counterexample**:
- If valid: Provide a **step-by-step proof** showing an argument for validity.
- If invalid: Build a **concrete model** M and identify an interval on it where the formula does not hold.
4. **Conclude**: Answer with uppercase "[VALID]" if the formula is valid or uppercase "[INVALID]" if it is not. No other responses are allowed.*

When *few shots* are used, three positive examples and three negative examples are extracted from a pool of pre-determined positive and negative examples containing 600 formulas, 300 of which are valid while the remaining ones are not, and randomly rotated for each individual problem.

We generated 1000 valid instances and 1000 non-valid ones, with length up to 139 symbols and modal depths up to 11, and submitted them in each of the 9 prompt configurations to each of the models. We used the following providers: DeepInfra, for Gemma 3 27b It, Qwen 3 32b, and Qwen 3 235b, NovitaAI for Gemma 3 27b It, and CentML for Llama 4.

The overview of the overall accuracies per model and per prompt configuration is reported in Tab. 3. The performances of each model and prompt configuration across progressively longer and progressively modally more complex is shown in Fig. 2. The first important

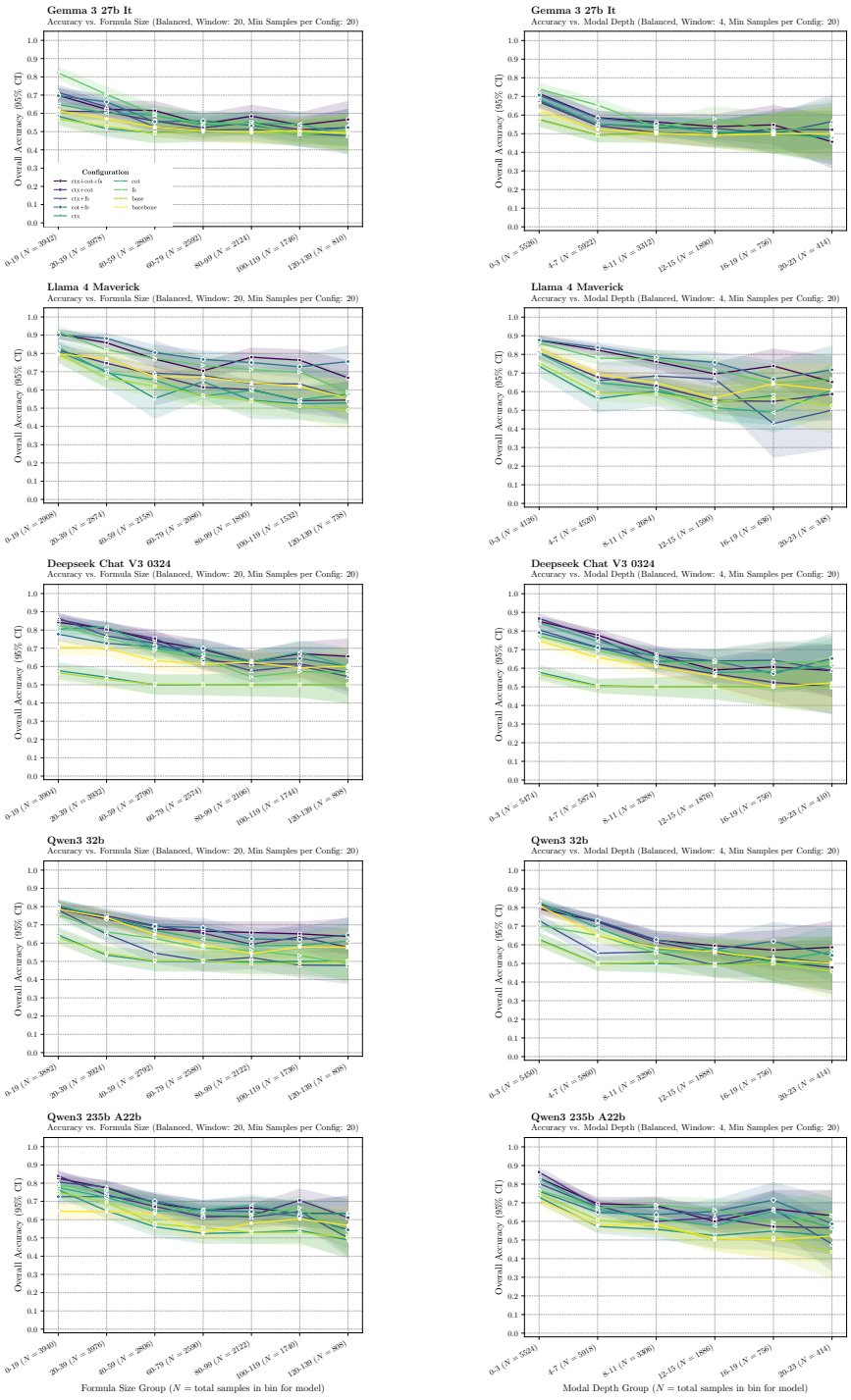


Figure 2 Accuracy per model, prompt configuration, and difficulty level, in terms of formula length (left hand side) and modal depth (right hand side).

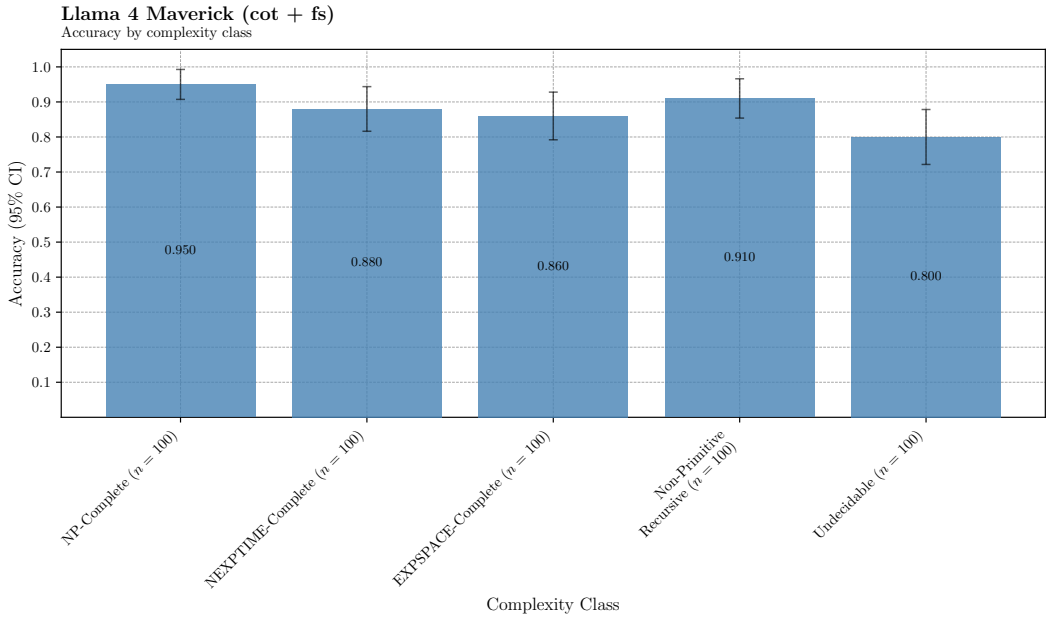


Figure 3 Accuracy of Llama 4 in the *ctx+fs* configuration for different complexity classes.

observation that can be drawn is that no model and no configuration reached more than 0.82 in overall accuracy; such value was achieved by Llama 4 in the *cot+fs* configuration. The overall accuracy in other models and configuration varies in a quite wide range, with a lower end of 0.52, which is essentially equivalent to the random answer. The models have behaved in very different ways to the different configurations. Those that have a lower overall accuracy across configurations, such as Gemma 3 27b It, seem to react positively to the progressively more detailed and precise information that is prompted from the *base* configuration up to the *ctx+cot+fs* one, although the improvement does not seem to be always linear. On the other hand, the ones with higher overall accuracy across configurations, such as Llama 4, seem not be too influenced by the type of prompt; Qwen 3 32b, in particular, presents an accuracy between 0.54 and 0.71 in all configurations, including *barebone*, indicating a close-to-null response from the instructions. All models have a predetermined strong bias towards answering that a formula is not valid (which in absolute terms is the most probable status of a random formula); DeepSeek Chat V3 showed the strongest bias: in two configurations, *ctx* and *base*, returned a true negative rate of 1.00, balanced by a true positive rate of 0.04 and 0.05, respectively. Adding *few shots* to the prompt, in general, slightly improves the results in almost every model.

Let us now analyse of the performances from the point of view of the intrinsic difficulty of the problem. The most evident phenomenon is the variability of the performances compared to the increasing hardness of the problem. Models, in general, exhibit the expected decrease of accuracy proportional to the length of the problem or its modal depth, but such decrease is not always clear. Thus, in some cases the worst performances do not correspond to the most difficult problems, such as in the case of Llama 4 Maverick and Qwen 3 235b, for several configurations.

Finally, in Fig. 3 we can see the result of a further experiment to assess the relationship between the ability of LLMs for interval temporal logic reasoning and the hardness of the problem in terms of computational complexity of the fragment that contains a formula. We

considered the top performing model(Llama 4) in the top performing configuration (*ctx+cot*), and devised a small dataset of 500 (small) formulas, 100 for each different computational class. As it can be seen, essentially no difference arises, despite the fact that the computational problem underlying such questions varies very much. The generally high performance is most probably due to formulas being short and with a low modal depth.

6 Conclusions

In this paper we considered the problem of benchmarking Large Language Models on their ability for formal logical reasoning, specifically interval temporal logical reasoning. Our results seem to indicate, quite reasonably, that statistical tools may not be the right solution for logical tasks; the fact that such tools are sometimes presented as representative of general intelligence, as well as the resonance that they have received in the recent past contributes to this confusion.

The high variability, the generally low accuracy, but most importantly the lack of consistency of the results is a clear indication of the unreliability of LLMs to perform logical reasoning on unseen problems. It is however of notice that some of models tested on our benchmark were capable, at least in some configurations, to correctly identify several valid and invalid formulas despite their high syntactical complexity, even if the tokenizer often produces syntactic mistakes such as merging double symbols (e.g., negation), useful for natural language but detrimental in this scenario.

References

1 L. Aceto, D. Della Monica, A. Ingólfsdóttir, A. Montanari, and G. Sciavicco. On the expressiveness of the interval logic of allen's relations over finite and discrete linear orders. In *Proc. of the 14th European Conference on Logics in Artificial Intelligence (JELIA)*, volume 8761 of *Lecture Notes in Computer Science*, pages 267–281. Springer, 2014. doi:10.1007/978-3-319-11558-0_19.

2 AI Insiders. Simple bench. https://simple-bench.com, 2024.

3 J. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983. doi:10.1145/182.358434.

4 F. Baader, D. Calvanese, D.L. McGuinness, and *others*, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

5 D. Bresolin, D. Della Monica, A. Montanari, P. Sala, and G. Sciavicco. Interval temporal logics over strongly discrete linear orders: Expressiveness and complexity. *Theor. Comput. Sci.*, 560:269–291, 2014. doi:10.1016/J.TCS.2014.03.033.

6 D. Bresolin, D. Della Monica, A. Montanari, and G. Sciavicco. A tableau system for right propositional neighborhood logic over finite linear orders: An implementation. In *Proc. of the 22th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX)*, volume 8123 of *LNCS*, pages 74–80. Springer, 2013. doi:10.1007/978-3-642-40537-2_8.

7 T.B. Brown, B. Mann, N. Ryder, and *others*. Language models are few-shot learners. In *Proc. of the 33rd Annual Confernce on Advances in Neural Information Processing Systems*, pages 1–25, 2020.

8 F. Chollet. On the measure of intelligence. *CoRR*, abs/1911.01547, 2019. arXiv:1911.01547.

9 P. Clark, O. Tafjord, and K. Richardson. Transformers as soft reasoners over language. In *Proc. of the 29th International Joint Conference on Artificial Intelligence*, pages 3882–3890, 2020.

10 K. Cobbe, V. Kosaraju, M. Bavarian, and *others*. Training verifiers to solve math word problems, 2021. arXiv:2110.14168.

11 V. Goranko, A. Montanari, P. Sala, and G. Sciavicco. A general tableau method for propositional interval temporal logics: Theory and implementation. *Journal of Applied Logics*, 4(3):305–330, 2006. doi:10.1016/J.JAL.2005.06.012.

12 V. Goranko, A. Montanari, and G. Sciavicco. Propositional interval neighborhood temporal logics. *Journal of Universal Computer Science*, 9(9):1137–1167, 2003. doi:10.3217/JUCS-009-09-1137.

13 V. Goranko, A. Montanari, and G. Sciavicco. A road map of interval temporal logics and duration calculi. *Journal of Applied Non-Classical Logics*, 14(1–2):9–54, 2004. doi:10.3166/JANCL.14.9-54.

14 Joseph Y. Halperns and Yoav Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, 1991. doi:10.1145/115234.115351.

15 S. Han, H. Schoelkopf, Y. Zhao, and *others*. FOLIO: natural language reasoning with first-order logic. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, 2024.

16 D. Hendrycks, C. Burns, S. Kadavath, and *others*. Measuring mathematical problem solving with the MATH dataset. In *Proc. of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks*, 2021.

17 T. Kojima, S. Shane Gu, M. Reid, and *others*. Large language models are zero-shot reasoners. In *Proc. of the 35th Annual Conference on Advances in Neural Information Processing Systems*, pages 1–15, 2022.

18 X. Lin, Q. Cao, Y. Huang, and *others*. ATG: benchmarking automated theorem generation for generative language models. In *Findings of the Association for Computational Linguistics*, pages 4465–4480. Association for Computational Linguistics, 2024. doi:10.18653/V1/2024.FINDINGS-NAACL.279.

19 E. Lucena-Sánchez, G. Sciavicco, and I.E Stan. Feature and language selection in temporal symbolic regression for interpretable air quality modelling. *Algorithms*, 14(3):76, 2021. doi:10.3390/A14030076.

20 F. Manzella, G. Pagliarini, G. Sciavicco, and I.E. Stan. The voice of COVID-19: breath and cough recording classification with temporal decision trees and random forests. *Artificial Intelligence in Medicine*, 137:102486, 2023. doi:10.1016/J.ARTMED.2022.102486.

21 T. Morishita, G. Morio, A. Yamaguchi, and *others*. Learning deductive reasoning from synthetic corpus based on formal logic. In *Proc. of the International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 25254–25274, 2023. URL: https://proceedings.mlr.press/v202/morishita23a.html.

22 E. Muñoz-Velasco, M. Pelegrín-Garcí, P. Sala, G. Sciavicco, and I. E. Stan. On coarser interval temporal logics. *Artificial Intelligence*, 266:1–26, 2019. doi:10.1016/J.ARTINT.2018.09.001.

23 T. Olausson, A. Gu, B. Lipkin, and *others*. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176. Association for Computational Linguistics, 2023. doi:10.18653/V1/2023.EMNLP-MAIN.313.

24 L. Pan, A. Albalak, X. Wang, and *others*. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics*, pages 3806–3824. Association for Computational Linguistics, 2023. doi:10.18653/V1/2023.FINDINGS-EMNLP.248.

25 M. Parmar, N. Patel, N. Varshney, and *others*. Logicbench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proc. of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 13679–13707, 2024.

26 D.A. Randell, Z. Cui, and A.G. Cohn. A spatial logic based on regions and connection. In *Proc. of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, pages 165–176. Morgan Kaufmann, 1992.

27 A. Saparov and H: He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *Proc. of the 11th International Conference on Learning Representations*, 2023.

28 A. Srivastava, A. Rastogi, A. Rao, and *others*. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

29 O. Tafjord, B. Dalvi, and P. Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics*, pages 3621–3634, 2021.

30 W. Tang and V. Belle. LTLBench: Towards benchmarks for evaluating temporal logic reasoning in large language models. *CoRR*, abs/2407.05434, 2024. doi:10.48550/arXiv.2407.05434.

31 J Tian, Y Li, W. Chen, and *others*. Diagnosing the first-order logical reasoning ability through logicNLI. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pages 3738–3747, 2021.

32 Y. Venema. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic*, 31(4):529–547, 1990. doi:10.1305/NDJFL/1093635589.

33 J. Wei, X. Wang, D. Schuurmans, and *others*. Chain-of-thought prompting elicits reasoning in large language models. In *Proc. of the 35th Annual Conference on Advances in Neural Information Processing Systems*, pages 1–14, 2022.

34 W. Zhong, R. Cui, Y. Guo, and *others*. AGIEval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics*, pages 2299–2314. Association for Computational Linguistics, 2024. doi:10.18653/V1/2024.FINDINGS-NAACL.149.

35 W. Zhong, S. Wang, D. Tang, and *others*. Analytical reasoning of text. In *Findings of the Association for Computational Linguistics*, pages 2306–2319, 2022.