JSONFOL: JSON 形式論理式による ニューロシンボリック推論

長谷川 遼　坂井 優介　上垣外 英剛　渡辺 太郎
奈良先端科学技術大学院大学
{hasegawa.ryo.hp5, sakai.yusuke.sr9, kamigaito.h, taro}@is.naist.jp

概要

言語モデルの基本的な能力である論理推論能力は，これまで様々な手法で評価および改善が図られてきた．その手法の 1 つであるニューロシンボリック手法では，最終的な判定を外部の定理証明器に任せることで，判定の性能および説明性を向上させる．ところが先行研究では定理証明器に関する専門的な知識が求められ，公平な評価が困難である．加えてエラー分析も論理式単位の荒い粒度のみと限定的だった．本研究では，まず公平な評価を実現するため，汎用的なスキーマである JSON で論理を扱う JSONFOL を提案し，その上で項単位での詳細なエラー分析を行った．その結果，対偶式の否定の位置や，式の局所的な複雑性などの局所的な要因が最終的な判定誤りの原因となることを明らかにした．また JSONFOL では最終的な判定を外部で行うため判定誤りを回避できることも判明した．

1　はじめに

大規模言語モデル（Large Language Model, LLM）は様々なタスクで高い性能を見せており [1, 2, 3, 4]，これに伴って幅広い言語能力の基礎となる論理推論能力の向上も期待されている [5, 6, 7]．しかしながら，否定を含む推論や命題数の多い推論など複雑な推論で性能が低下する [8, 9, 10]，推論過程がブラックボックス的で説明性が低い等の課題もある．

これを解決する方針の 1 つにニューロシンボリック手法がある [11, 12, 13]．論理推論タスクの最終的な正誤判定を外部の定理証明器に任せ，LLM には論理式のコードのみを生成させることで，既存手法よりも高い性能を記録している．

ところがニューロシンボリック手法にも課題がある．LLM には定理証明器に関する専門的な知識が要求されるため，LLM の論理推論能力の公平な評価ができない．加えて既存研究では否定の有無や命題数など論理式単位での比較的荒い分析しか行われず，より局所的な部分論理式による影響の評価までは至っていない．

本研究ではまず，公平な評価が可能なニューロシンボリック手法として JSONFOL を提案する．JSONFOL は汎用的なスキーマである JSON によって一階述語論理を表現できるため，言語モデルに専門的な知識の理解を要求せず論理推論能力を公平に評価できる．そして粒度の細かい分析のために，部分論理式単位での差分データを豊富に含む AAC [14] をデータセットとして使い分析を行った．実験の結果，通常の推論では対偶式の否定の位置や局所的な複雑性など，一部の部分論理式が最終的な判定の誤りの原因となると判明した．加えて JSONFOL の，最終的な判定をモデル内部では実行しない構造が，部分論理式による誤りの改善に貢献することも明らかになった．

2　関連研究

プロンプトによる推論能力の改善　モデルの論理推論能力を向上させるアプローチの 1 つにプロンプトがあり，Chain-of-Thought (CoT) [5] が一般的に使われる．CoT ではモデルの入出力の例に加え，推論プロセス例もプロンプトに与える．また “Let’s think step by step” のように推論を促すフレーズを与えることで，推論プロセスを明示せずとも推論能力の向上が可能なことが知られている [6]．一方で課題として，複雑な推論では推論プロセスが最終的な出力に反映されないことがある [15, 16]．

ニューロシンボリック手法　推論プロセスを明示するために言語モデルと記号推論を組み合わせたニューロシンボリック手法はいくつか存在する．LINC [12] では 1 つの入力に対して複数回推論を行う．LLM には外部の自動定理証明器である

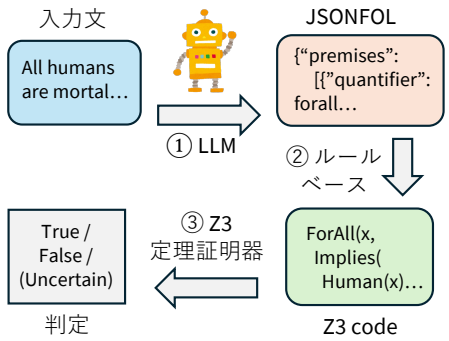

図 1: JSONFOL による論理推論の処理の流れ

Prover9 [17] のコードを出力させ，多数決で最終的な判定を行う．LoRP [13] では Prolog が使用されている．いずれの手法も LLM に対して定理証明器に関する専門的な知識の運用を求めており，モデル間の公平な能力比較という用途は想定されていない．

3 提案手法：JSONFOL

JSONFOL は JSON による一階述語論理の表現手法である．図 1 に JSONFOL を使った論理推論タスクの流れを示す．判定までに 3 段階の処理を行う．

例えば入力文として“All humans are mortal. Socrates is human. Therefore, Socrates is mortal.”（論理式としては $\forall x\, human(x) \rightarrow mortal(x), human(Socrates) \Rightarrow mortal(Socrates)$ に相当）が与えられた場合を考える．まずは LLM は入力文を以下のような JSONFOL に変換する．

```
[JSONFOL]
{"premises":
  [{"quantifier": "forall",
      "variable": "x",
      "formula":
        {"if": {"predicate": "human", "args":
            [{"var": "x"}]},
         "then": {"predicate": "mortal", "args":
            [{ "var": "x"}]}}},
    {"predicate": "mortal", "args": [{"const":
        "Socrates"}]}],
 "conclusion": {"predicate": "mortal", "args":
     [{"const": "Socrates"}]}}
```

JSONFOL はキーとして一階述語論理の各要素の名前を記載する．例えば定数（"const"）や変数（"var"），項（"args"），述語（"predicate"），論理演算子（"if"と"then"の組, "not", "and", "or"など），量化子（"quantifier"と"variable"の組）が使用される．量化子は全称量化子を"forall"で，存在量化子を"exists"で表現する．値としては定数（"Socrates"など）や変数（"x"など）を格納する他，句や節を表す JSON 構造そのものを再帰的に格納することもできる．

次に，出力した JSONFOL に対してルールベースの後処理を適用し，自動定理証明器である Z3 [18] のコードに変換する．

最後に Z3 で判定を行い，真理値を出力する．

他のニューロシンボリック手法と異なり，LLM に対して定理証明器のコードに関する知識を直接求めないため，論理推論能力を公平に比較できる．

4 実験設定

データセット 本研究では，データセットとして AAC を使用した．AAC は一階述語論理の論理式に基づく合成データセットである．8 種類の論理スキーマと 4 種類の変形からなる 32 のグループに分かれ，各グループに 1-3 種類の論理式が属し，計 71 論理式を含む．自然文は各論理式に対応したテンプレートに沿って生成される．タスクは True/False の二値判定であり，False のサンプルは論理式の最終項の否定を反転させて生成している．詳細は付録 A で述べる．本実験では，AAC の評価用データから各論理式について正例 10 件と負例 10 件，合わせて 20 件をランダムに抽出した（Qwen3 のみ計算時間の都合上，正例 2 件と負例 2 件）．71 論理式全てについて，合計 1420 件（Qwen3 は 284 件）の分析を行った．

言語モデル GPT-4o-mini, GPT-4o [1], GPT-5 [2], Gemini-2.5-Flash [3], Qwen3 （Qwen3-30B-A3B-Thinking-2507） [4] を分析対象とした．GPT-5, Gemini-2.5-Flash では，Reasoning トークンの設定を medium とした．実装には OpenAI API, Gemini API, Hugging Face Transformers [19] を使用した．

プロンプト 本実験では判定のみ（naive），Chain-of-Thought (CoT) [5, 6], JSONFOL の 3 種のプロンプトを使用した．naive では LLM に最終的な判定（True または False）のみを，CoT では判定と推論過程を出力させた．JSONFOL では JSONFOL の仕様をプロンプトに記入し，入力文を仕様通りに変換するよう指示した．得られた JSONFOL 構造はルールベースの後処理で Z3 のコードに変換し，Z3 の定理証明器で判定した．

評価指標 使用した指標を表 1 に示す．JSONFOL では Overal accuracy を Success rate と Conditional accuracy に分け，前者で指示追従能力，後者で指示通り出力できた JSONFOL の正しさを測った．naive および CoT では Overall accuracy のみを用いた．

表 1: 実験に使用した指標

Overall accuracy	Success rate	Conditional accuracy
正解数 / 全サンプル数	出力に成功した数 / 全サンプル	正解数 / 出力に成功した数

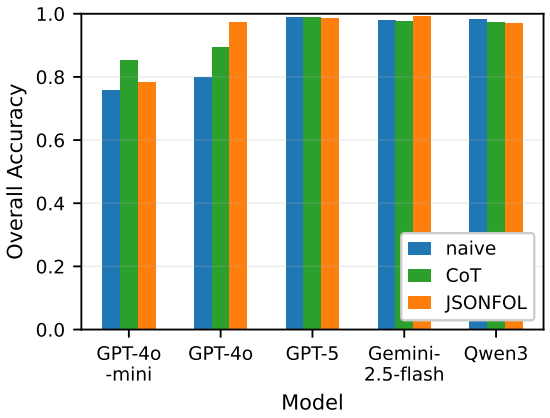

図 2: 言語モデル・プロンプト別の Overall accuracy

5 実験結果と考察

5.1 全体の傾向

表 2 で overall accuracy を示す．論理推論モデルである GPT-5, Gemini-2.5, Qwen3 では全てのプロンプトで accuracy が 1 に近かった．他モデルでも JSONFOL は naive より高く，特に GPT-4o では論理推論モデルと同等程度に高かった．以上より JSONFOL はモデルの評価に耐えうる手法であることが明らかになった．

表 2 に JSONFOL の success rate と conditional accuracy を示す．GPT-4o-mini 以外の 4 モデルで両者とも 1 に近く，GPT-4o-mini においては conditional accuracy が success rate より高かった．JSONFOL のエラーは指示通りでない出力によるものが主であり，仕様通りに出力された JSONFOL は論理的な破綻が少ないことが分かった．

5.2 エラー分析

この章では部分論理式単位のエラー分析を行う．JSONFOL の指標には Contitional accuracy を使う．

対偶 図 3 に以下の 4 種類の対偶式の accuracy を示す．対偶の式とラベル名の対応は以下の通り．

base schema:

$\forall x\, F(x) \rightarrow \neg G(x) \Rightarrow \forall x\, G(x) \rightarrow \neg F(x)$

negation variant 1:

$\forall x\, \neg F(x) \rightarrow G(x) \Rightarrow \forall x\, \neg G(x) \rightarrow F(x)$

表 2: JSONFOL における Success rate と conditional accuracy

Models	Success rate	Conditional accuracy
GPT-4o-mini	0.848	0.916
GPT-4o	0.990	0.982
GPT-5	0.996	0.989
Gemini-2.5-flash	0.995	0.996
Qwen3	0.971	1.000

negation variant 2:

$\forall x\, \neg F(x) \rightarrow \neg G(x) \Rightarrow \forall x\, G(x) \rightarrow F(x)$

negation variant 3:

$\forall x\, F(x) \rightarrow G(x) \Rightarrow \forall x\, \neg G(x) \rightarrow \neg F(x)$

これらの対偶式は否定の位置のみが異なり，項数や式数など式全体の難しさは同等である．にも変わらず，naive と CoT では negation variant 1 と 2 が他の式よりも明確に accuracy が低い．一方で JSONFOL ではほぼ全条件で naive と CoT より accuracy が高い．

表 3 では，negation variant 1 を GPT-4o で判定した際の confusion matrix を示す．Naive および CoT では，False が正答である式をほぼ全て True と判定している．これは GPT-4o が論理式中の誤りを見逃す傾向にあることを示す．

CoT での GPT-4o の典型的な誤りを以下に示す．

```
[input]
Consider the following argument: If someone is
    not a contemporary of Spinosaurus, then
    they are a predator of Triceratops. Thus,
    if someone is not a predator of
    Triceratops, then they are not a
    contemporary of Spinosaurus.

[GPT-4o CoT output]
[...] can be written in logical form as: \(
    \neg C(x) \rightarrow P(x) \)[...].
The conclusion [...] can be written as: \( \neg
    P(x) \rightarrow \neg C(x) \).
The conclusion is the contrapositive of the
    given statement. [...] the conclusion is
    true given the premise.
{Answer: true}
```

この例では対偶は成り立たず，正答は False となる．GPT-4o の CoT では個々の式の理解は正しい．ところが最終的な判定時にモデルが典型的な対偶であると思い込み，True と誤答している．この誤りは一部の対偶式でのみ出現するため，訓練データ中に特定の対偶式に対するバイアスがある可能性がある．また判定時の誤りは，判定をモデル外部で行う

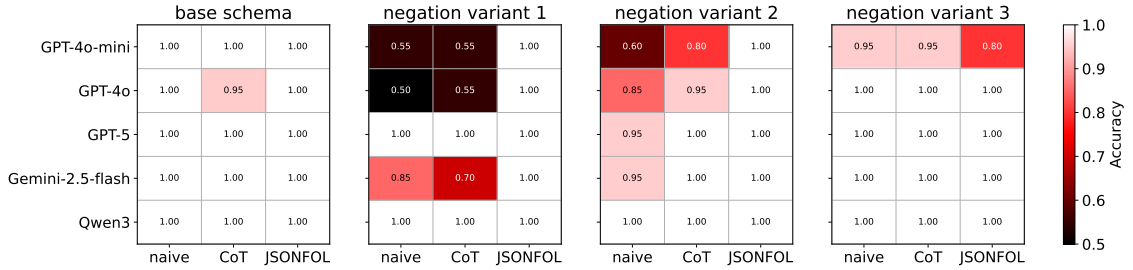

図 3: 否定の位置が異なる対偶の式の accuracy

表 3: 対偶の negation variant 1 式における GPT-4o での Confusion matrix

naive	True	False
True	10	0
False	10	0

CoT	True	False
True	10	0
False	9	1

JSONFOL	True	False
True	10	0
False	0	10

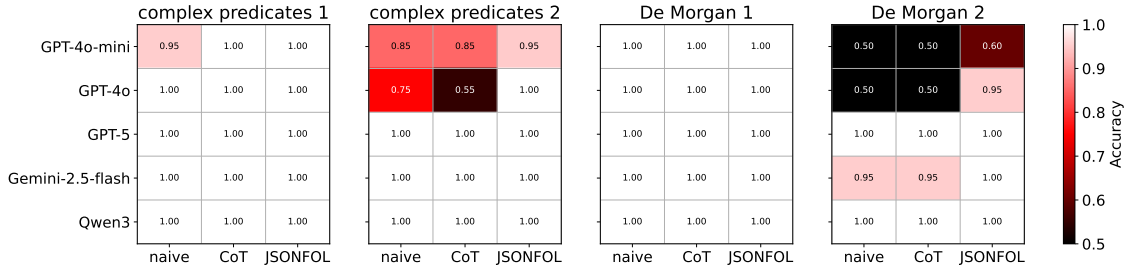

図 4: 仮説後件の項数が異なる論理式の accuracy

JSONFOL では回避できることも明確になった．

局所的な複雑性　AAC では，True データを仮説後件の最終項の否定を反転させることで False データを生成している．そのため本研究では，局所的な複雑性の重要な例として仮説後件の項数を分析対象とした．図 4 は，仮説後件の項数が 2 および 1 である 4 つの式の accuracy を示す．この 4 式の論理式全体の難しさ（式数および項数）は等しい．

complex predicates 1 (仮説後件の項数 1):
$\forall x\,(F(x) \land I(x)) \to G(x), \quad \forall x\,G(x) \to H(x)$
$\Rightarrow \quad \forall x\,(F(x) \land I(x)) \to H(x)$

complex predicates 2 (仮説後件の項数 2):
$\forall x\,(F(x) \to G(x)), \quad \forall x\,G(x) \to (H(x) \lor I(x))$
$\Rightarrow \quad \forall x\,F(x) \to (H(x) \lor I(x))$

De Morgan 1 (仮説後件の項数 1):
$\forall x\,(\neg F(x) \land \neg I(x)) \to G(x), \quad \forall x\,G(x) \to H(x)$
$\Rightarrow \quad \forall x\,\neg(F(x) \lor I(x)) \to H(x)$

De Morgan 2 (仮説後件の項数 2):
$\forall x\,\neg F(x) \to G(x), \quad \forall x\,G(x) \to (\neg H(x) \lor \neg I(x))$
$\Rightarrow \quad \forall x\,G(x) \to \neg(H(x) \land I(x))$

GPT-4o-mini および GPT-4o において，仮説後件の項数が 2 である complex predicates 2 と De Morgan 2 は，naive や CoT で accuracy が他より明確に低い accuracy．一方で JSONFOL は全条件で naive および CoT よりも accuracy が高い．

以上より，論理式全体の複雑さが等しくても，正誤の根拠となる項に関連する項の数が多ければ，判定を誤りやすい傾向があることが分かった．また対偶と同様，JSONFOL 等で判定を言語モデル外部で行うことで誤りを回避できることも判明した．この傾向は hypothetical syllogism 1 に限らず AAC 全体で一貫していた（付録 B）．

6　結論

言語モデルの論理推論能力の公平な比較のため，我々は汎用的なデータ構造である JSON で論理を扱う JSONFOL を提案した．JSONFOL と naive なプロンプトや CoT を比較して一階述語論理タスクを実施し，部分論理式単位での詳細なエラー分析を行った．その結果，naive なプロンプトや CoT では，対偶の式の否定の位置・局所的な複雑性などの部分的な要因が LLM の最終的な判定の誤りを誘発すること，JSONFOL では判定を外部で行うため誤りを回避できることが判明した．

謝辞

本研究は JSPS 科研費 JP23H03458，JP25K24369 の助成を受けたものです．

参考文献

[1] OpenAI, Josh Achiam, Steven Adler, et al. Gpt-4 technical report, 2024.

[2] OpenAI. Gpt-5 system card., 2025.

[3] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025.

[4] An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report, 2025.

[5] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, **Advances in Neural Information Processing Systems**, Vol. 35, pp. 24824–24837. Curran Associates, Inc., 2022.

[6] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In **Proceedings of the 36th International Conference on Neural Information Processing Systems**, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.

[7] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In **The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023**. OpenReview.net, 2023.

[8] Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. LogicBench: Towards systematic evaluation of logical reasoning ability of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 13679–13707, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[9] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: limits of transformers on compositionality. In **Proceedings of the 37th International Conference on Neural Information Processing Systems**, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[10] Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Seyed Mehran Kazemi, Najoung Kim, and He He. Testing the general deductive reasoning capacity of large language models using ood examples. In **Proceedings of the 37th International Conference on Neural Information Processing Systems**, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[11] Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Findings of the Association for Computational Linguistics: EMNLP 2023**, pp. 3806–3824, Singapore, December 2023. Association for Computational Linguistics.

[12] Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 5153–5176, Singapore, December 2023. Association for Computational Linguistics.

[13] Zhengkun Di, Chaoli Zhang, Hongtao Lv, Lizhen Cui, and Lei Liu. Lorp: Llm-based logical reasoning via prolog. **Knowledge-Based Systems**, Vol. 327, p. 114140, 2025.

[14] Gregor Betz, Christian Voigt, and Kyle Richardson. Critical thinking for language models. In Sina Zarrieß, Johan Bos, Rik van Noord, and Lasha Abzianidze, editors, **Proceedings of the 14th International Conference on Computational Semantics (IWCS)**, pp. 63–75, Groningen, The Netherlands (online), June 2021. Association for Computational Linguistics.

[15] Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. **Advances in Neural Information Processing Systems**, Vol. 36, pp. 74952–74965, 2023.

[16] Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. **arXiv preprint arXiv:2307.13702**, 2023.

[17] William McCune. Release of prover9. In **Mile high conference on quasigroups, loops and nonassociative systems, Denver, Colorado**, 2005.

[18] Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In **International conference on Tools and Algorithms for the Construction and Analysis of Systems**, pp. 337–340. Springer, 2008.

[19] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 38–45, Online, October 2020. Association for Computational Linguistics.

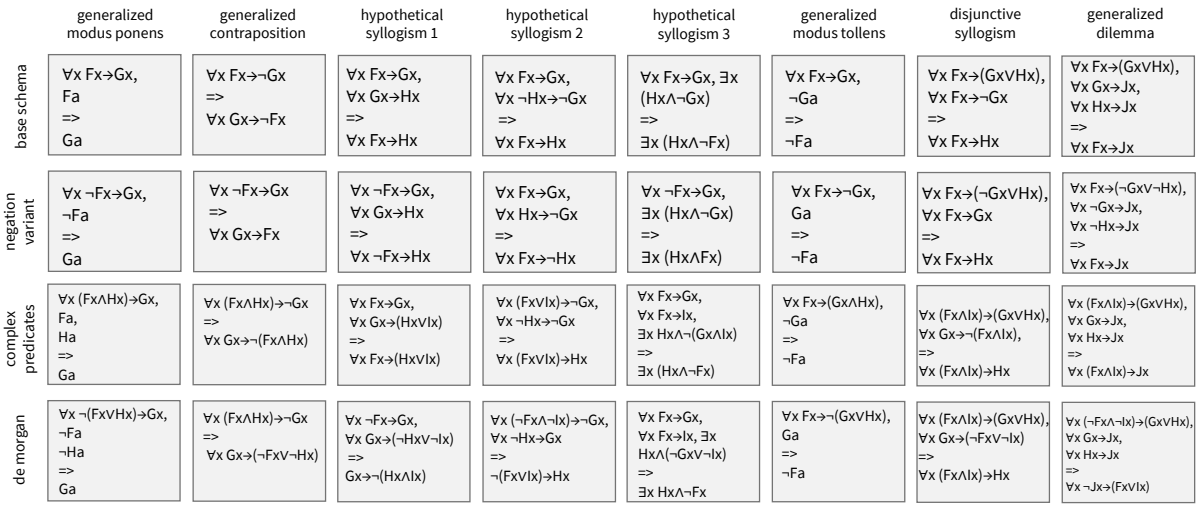

図 5: AAC の論理式スキーマと変形の一覧

A　データセット AAC の詳細

図 5 に AAC の論理式の大まかな分類を示す．AAC では generalized modus ponens, generalized contraposition, hypothetical syllogism 1, hypothetical syllogism 2, hypothetical syllogism 3, generalized modus tollens, disjunctive syllogism, generalized dilemma8 つの論理式スキーマが用意されている．各スキーマに対して base schema, negation variant, complex predicates, De Morgan の 4 種の変形が施され，計 32 のグループが形成される．Negation variant は base schema の否定の反転，complex predicates は base schema への項の追加，De Morgan は complex predicates への De Morgan 則の適用を意味する．変形は複数通り考えられることから，各グループには 1 から 3 種の論理式が含まれる．このように，AAC では論理式を体系的に整理している．

B　仮説後件の項数の詳細な分析

この章では，論理式の仮説後件の項数と Accuracy の関係を AAC データセット全体で分析する．表 4 において，Two-terms は仮説後件の項数が 2 の論理式を表す．Two-terms の式は論理式スキーマとしては generalized contraposition, hypothetical syllogism 1–3, disjunctive syllogism および generalized dilemma に属している．また変形としては complex predicates および De Morgan に属している．一方で，One-term は仮説後件の項数が 1 の論理式のうち，論理式全体の複雑さが Two-terms と同等なもの，則ち論理式スキーマと変形が同じ式を表している．

表 4: 仮説後件の項数が 1 および 2 である論理式全体の Accuracy

Two-terms	naive	CoT	JSONFOL
GPT-4o-mini	0.595	0.570	0.721
GPT-4o	0.535	0.545	0.922
GPT-5	0.935	0.935	0.920
Gemini-2.5-flash	0.920	0.914	0.985
Qwen3	0.875	0.875	0.975
One-term	**naive**	**CoT**	**JSONFOL**
GPT-4o-mini	0.821	0.946	0.985
GPT-4o	0.853	0.989	0.996
GPT-5	1.000	1.000	1.000
Gemini-2.5-flash	0.996	1.000	1.000
Qwen3	1.000	1.000	1.000

表 4 は Two-terms と One-term の accuracy を示す．論理式全体の難易度は同等であるにも関わらず，Naive と CoT ではすべてのモデルにおいて，One-term と比較して Two-terms の accuracy が明確に低くなっている．したがって，仮説後件の項数が多い，すなわち True/False を決定する重要な項の周辺が複雑である場合に，モデルが判定を誤りやすい，という傾向が共通していると明らかになった．加えて，判定を外部の定理証明器で行う JSONFOL では誤りを回避できるという傾向の一貫性も確認された．