Minimal Free Resolution Guided Adaptive Tree Reasoning

Dezhao Tang[1,2,3], **Meihan Liu**[1,2,3], **Yulai Tong**[1,2,3], **Guan Yuan**[1,2,3], **Qiuyan Yan**[1,2,3*]
[1]School of Computer Science and Technology / School of Artificial Intelligence,
China University of Mining and Technology
[2]Jiangsu Provincial Industrial Technology Engineering Center for Intelligent Sensing
and Emergency IoT in Underground Space
[3]Mine Digitization Engineering Research Center of the Ministry of Education
{tb24170004a41, damei, yl_t, yuanguan, yanqy}@cumt.edu.cn

Abstract

Dynamic reasoning trees can help large language models solve complex tasks by explicitly structuring intermediate decisions. However, existing approaches often rely on manually specified subproblems or predefined decomposition patterns, which limits the effectiveness of reasoning and generalization. To solve this problem, we propose SyRA, a hierarchical reasoning framework based on MFR theory that supports the construction of adaptive reasoning trees and reliable error correction within a single LLM. Specifically, SyRA focuses on reasoning-tree construction, dynamically controlling branching and expansion using MFR principles to enable informative, non-redundant subproblem decomposition. In addition, it introduces a residual backtracking mechanism for adaptive cross-layer error correction, allowing the model to revise earlier reasoning decisions based on downstream feedback. Across eight reasoning benchmarks, SyRA significantly reduces logical errors and improves reasoning accuracy, while achieving a better balance between accuracy and reasoning time than the Chain-of-Thought, Decompose–Analyze–Rethink and Tree-of-Thought. Our code and dataset are available at https://github.com/Tim798-art/SyRA/tree/main/SyRA.

1 Introduction

Large Language Models (LLMs), originally designed for text generation, have shown great potential for a variety of reasoning tasks (Zhang et al., 2025; Xiao et al., 2024; Manas et al., 2024). Surprisingly, this potential still relies on standard autoregressive generation (Cheng et al., 2025; Bao et al., 2025; Pulakurthi et al., 2025). However, this linear generation process behaves like a black box, making it difficult to support multi-hypothesis exploration and dynamic correction in complex scenarios(Cheng et al., 2024). Therefore, establishing a reasoning paradigm that supports explicit branching, verification, and error correction is crucial for enhancing the structured reasoning capabilities of LLMs and enabling reliable real-world decision-making applications(Ma et al., 2025).

Against this backdrop, researchers proposed the ToT reasoning framework, which enables LLMs to exhibit more human-like reasoning abilities in complex reasoning tasks (Wu et al., 2025). As a highly structured and verifiable domain, mathematical reasoning serves as a common testbed for evaluating such methods(Setlur et al., 2024; Satpute et al., 2024; Zhang et al., 2024). From Figure 1, guided by a pre-built decomposition example pool and a threshold-based trigger, the traditional adaptive tree method uses LLM reasoning to build an adaptive tree with intermediate nodes. It then adaptively expands, terminates, and backtracks at each node to identify a complete solution path, where the blue path denotes the decomposition process and the green path denotes the backtracking process.

Despite mitigating ToT's rigidity, adaptive tree reasoning can still limit problem-solving performance(Besta et al., 2024). **First**, tree growth depends on a pre-built decomposition example pool and a manually specified threshold, which can undermine the effectiveness and generalizability of reasoning(Wu et al., 2024). When the pool lacks an appropriate decomposition, the model may adopt a mismatched template and consequently produce incorrect conclusions. Due to the lack of a decomposition example for q_2 in the pool, the decomposition omits the subquestion "How many ... muffins?" in Figure 1(a). **Second**, errors at lower-level nodes can directly affect the updating of the parent node's reasoning results(Yang et al., 2025). If the reasoning for q_2 deviates, the model cannot effectively leverage information from deeper child nodes to update the root-level conclusion, ultimately leading to incorrect outputs (e.g., '34').

To address these issues, we propose an adaptive

*Qiuyan Yan is the corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 46507–46522

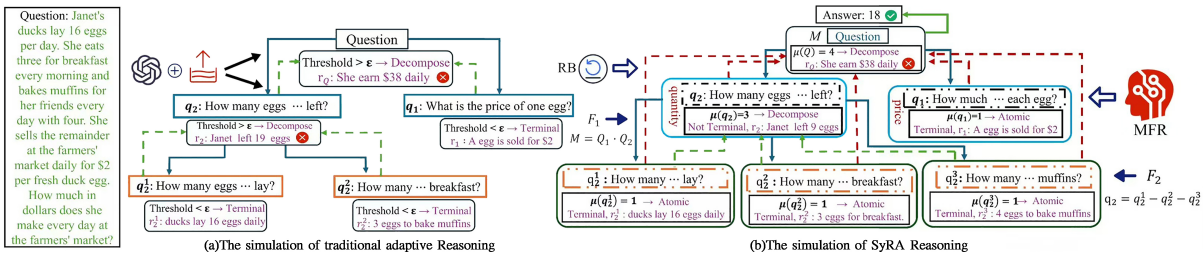

Figure 1: Comparison of ToT variants.

tree-based reasoning framework, **Sy**zygy-driven **R**easoning **A**dapter(**SyRA**) (Gimenez et al., 2010), which guides LLMs toward human-like reasoning on complex problems. Here, syzygy denotes the structured dependencies among subproblems rather than mere sequential relations between reasoning steps. Based on Minimal Free Resolution (MFR), we formulate tree-structured problem decomposition in a hierarchical free-representation space, enabling a more structured organization of reasoning and more coherent problem solving. As illustrated in Figure 1(b), the original problem is first decomposed into two subproblems, q_1 (price) and q_2 (quantity). The subproblem q_2 is further decomposed into q_2^1, q_2^2, and q_2^3, corresponding to production, breakfast, and consumption. These subproblems are solved sequentially and their results are used to update q_2 and then the root problem q_{root}, ultimately yielding the final answer ('18'). From a structural perspective, at layer F_1, the generated subproblems q_1 and q_2 satisfy $M = q_1 \times q_2$, where their product recovers the original profit. Similarly, the decomposition of q_2 as an independent problem module follows a similar compositional relation.

To realize such a human-like framework for solving complex problems, we proceed as follows. **(i)** Based on free generating elements, SyRA decomposes the problem into simpler subproblems at child nodes (blue arrows). **(ii)** SyRA decides whether further decomposition is required based on the number of free generating elements (lower boxes of each child node). **(iii)** Inspired by residual learning (He et al., 2015), SyRA introduces a cross-level residual backtracking strategy (RB): when a node receives a low consistency score, SyRA aggregates relevant information from its all descendant nodes to refine the result of the reasoning of the node (red and green arrows). So, our contributions are as follows:

• **Theory:** We prove that tree-structured problem decomposition can be mapped to MFR's hierarchical free-representation space and meet its applicability conditions, thereby addressing the lack of effective heuristics for this task.

• **Method:** We propose SyRA, a human-like adaptive reasoning framework that incorporates MFR's minimality, exactness, and syzygy mechanisms to enable coherent tree-structured reasoning and adaptive cross-layer error correction.

• **Results:** Extensive experiments show that SyRA outperforms existing methods in closed reasoning settings and remains strong in open-domain factual reasoning without additional knowledge.

2 Related Work

Early research relied primarily on linguistic rules, using manually crafted decomposition strategies based on syntactic analysis to break complex questions into direct solvable queries(Kalyanpur et al., 2012). However, these approaches relied heavily on manually designed rules and were difficult to apply to a wider range of tasks.

As neural networks became widely adopted in natural language processing, question decomposition was increasingly modeled as a supervised sequence-generation or sequence-labeling task (Zhang et al., 2019). The core idea is to train models to generate executable intermediate structures to solve complex questions. For example, question decomposition can be formulated as a sequence generation task that decomposes a complex question into multiple subquestions (Min et al., 2019). In this setting, the model generates subquestions without complex structured annotations. By directly generating subqueries for downstream QA systems, this approach is intuitive and easy to integrate. In contrast, BREAK adopts a structured paradigm by collecting complete questions' decomposition meaning representation annotations to explicitly encode reasoning steps(Wolfson et al., 2020). However, limited high-quality annotations remain a major bottleneck for both performance and generalization.

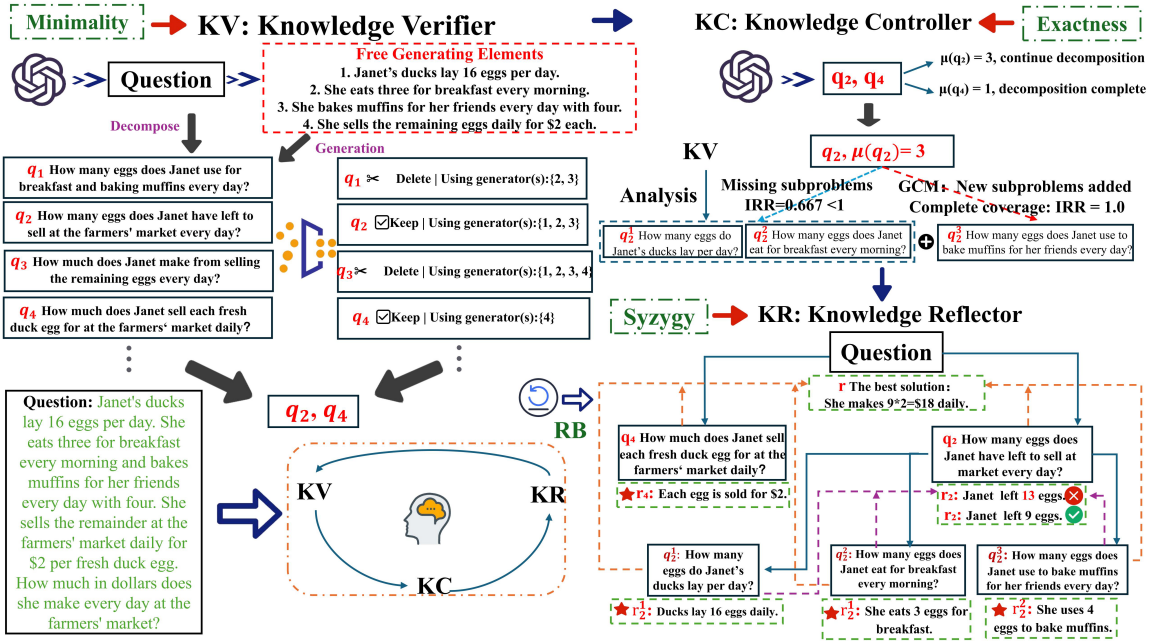

Figure 2: A demonstration of the SyRA cycle.

With the continued development of LLMs, researchers have increasingly used them for question decomposition (Chen et al., 2022). For example, ToT prompts LLMs to search for intermediate reasoning steps, gradually constructing a complete reasoning process (Yao et al., 2023). In addition, Least-to-Most Prompting guides LLMs to decompose questions into a sequence of simpler subproblems using only a few examples (Zhou et al., 2022). Decompose–Analyze–Rethink (DeAR) iteratively decomposes problems, analyzes subquestions, and refines solutions by rethinking higher-level reasoning with lower-level feedback (Xue et al., 2024).These carefully designed prompting methods have made question decomposition a key component of LLMs for complex reasoning.

3 SyRA Model

Given a question Q("Janet's... market?"), SyRA produces a reasoning $R = (r_1, r_2, \ldots, r_k)$ and the corresponding answer (e.g., '18'). Our goal is to design an MFR-guided adaptive tree framework that ensures full tree growth, enables cross-layer reflection, and produces the final answer. In Figure 2, SyRA builds the KV–KC–KR cycle for human-like progressive reasoning on complex problems.

KV: The pretrained T5 model first extracts from the problem a minimal set of solution-relevant free generating elements, which capture the key information required for solving it. The subproblems generated from the problem and those derived from the free generating elements are then merged to form the subproblem set $\{q_1, \cdots, q_4\}$. A hybrid filtering mechanism based on minimality is then applied to the elements of these subproblems, yielding the final subproblems q_2 and q_4.

KC: Subproblems whose number of elements falls below a threshold are treated as leaf nodes; otherwise, they are further decomposed. If the upper-to-lower element ratio is less than 1 ($\mathrm{IRR}(q_2) = 0.667$), we use GCM to generate the corresponding subproblem q_3^2 from the missing elements. This process ensures complete information transfer across layers and thereby guarantees exactness.

KR: Each subproblem is a tree node, and the LLM answers it with the original problem as context. If a node's reasoning result receives a low consistency score, KR uses an RB strategy based on syzygy to aggregate information from all its descendant nodes, select the most reliable reasoning path, and refine the node's reasoning.

3.1 Knowledge Verifier Construction

This section proposes a Knowledge Verifier (KV), guided by the minimality, to enable efficient and controllable hierarchical subproblem generation.

To bridge the problem with the affine representation $(S, I(Q), M(Q))$, we train a constraint discriminator $F_{\text{gen}}(\cdot)$ under learning-based heuristic supervision to extract free generating elements as the core constraints in $I(Q)$ (Section 3.4 and Ap-

pendix A.1). So, we view $M(Q)$ as a structural representation implicitly induced by LLMs during tree-based reasoning. From Appendix A.2, we extract the set of free generating elements $E(Q)$ and denote its cardinality by γ.

We adopt progressive refinement to improve subproblem coverage at each level. Given Q_i, P_1 guides M_{decomp} to explore semantics and generate a higher-coverage candidate subproblem set:

$$Q_i^{(1)} = M_{\text{decomp}}(Q_i, P_1) = \{q_1, \ldots, q_m\} \quad (1)$$

Where $Q_i^{(1)}$ is the set of first-layer subproblems decomposed from Q_i, and q_m is a subproblem in this set. Subsequently, using the set of $E(Q_i)$ identified in Q_i, we use the prompt P_2 to guide M_{gen} in generating a more targeted set of subproblems:

$$Q_i'^{(1)} = M_{\text{gen}}(Q_i, E(Q_i), P_2) = \{q_1', \ldots, q_n'\} \quad (2)$$

Finally, the two subproblem types are combined to form the final set of subproblems for this level:

$$Q_i^{*(1)} = Q_i^{(1)} \cup Q_i'^{(1)}. \quad (3)$$

We enforce minimality to keep the subproblem set compact. By integrating the LLM-based semantic judgment with a hybrid filtering mechanism grounded in free generating elements, we achieve dynamic subproblem pruning that is both controllable and interpretable. Therefore, the verifier selects the retained subproblem set Q_i^* according to the following rules at each layer:

(1) **Question Pruning**: LLMs often carry over the final query of the original problem into generated subproblems, weakening subsequent reasoning. In Appendix A.3, we use constraint prompts (e.g., "remove at most one") to determine whether a generated subproblem is semantically equivalent to this final query. We then greedily prune subproblems that are semantically equivalent or highly similar to this final query.

(2) **Redundancy Compression**: The verifier compares free generating elements of same-layer subproblems and detects redundancy via set inclusion. If the elements of the current subproblem are a subset of those of a retained subproblem at the same layer (i.e., $E(q_i) \subseteq E(q_j)$), the current subproblem is considered redundant and discarded.

(3) **Subproblem Retention**: If the free generating elements of two subproblems are independent (i.e., their intersection is empty), or if their intersection is non-empty but does not yield a full inclusion (i.e., $E(q_i) \cap E(q_j) \neq E(q_i)$ and $E(q_i) \cap E(q_j) \neq E(q_j)$), then both subproblems are deemed valid and retained.

3.2 Knowledge Controller Construction

This section proposes a Knowledge Controller (KC), guided by exactness, to determine when to decompose and to ensure the complete inter-layer information propagation.

In the i-th layer of the adaptive tree, let the retained subproblem set be Q_i^*. Any subproblem q_i^j that satisfies the threshold $\Upsilon_i^j \geq \Upsilon_{\min}$ is added to the pending set of decompositions $\widehat{Q}_i^*$.

This iterative process continues until the termination condition is satisfied: the elements of all subproblems do not exceed a predefined threshold, i.e., $\max \Upsilon_i^j \leq \Upsilon_{\min}$.

Let $E(q_i^j)$ denote the set of free generating elements of the subproblem $q_i^j \in Q_i^*$, and their union at this layer is given by:

$$G_i = \bigcup_{q_i^j \in Q_i^*} E(q_i^j) \quad (4)$$

To assess whether the information of a parent node is retained in its subproblems, we define the Inter-layer Retention Rate (IRR) based on minimal free generating elements to quantify the completeness of the decomposition:

$$R_i = \frac{|G_{i-1} \cap G_i|}{|G_{i-1}| + \varepsilon} \quad (5)$$

where $\varepsilon > 0$ is a numerical stabilizer. If $R_i = 1$, the free generating elements from the previous layer are fully covered by the current layer; otherwise, the uncovered elements trigger the Global Coverage Mechanism (GCM), which constructs new subproblems from missing element information to supplement the uncovered nodes:

$$q_i^{\text{new}} = M_{\text{gen}}(Q_i, E(Q_i), P_3) \quad (6)$$

3.3 Knowledge Reflector Construction

Guided by syzygy, we introduce a Knowledge Reflector (KR) with a Residual Backtracking (RB) strategy to improve reasoning accuracy.

The filtered subproblems in layer i are modeled as structured reasoning nodes:

$$n_i^j = (\, q_i^j,\ r_i^j,\ s_i^j \,) \quad (7)$$

Where q_i^j denotes the subproblem to be solved, r_i^j is generated by M_{Re} under prompt P_4, s_i^j is the consistency score computed by M_{Sc} under prompt

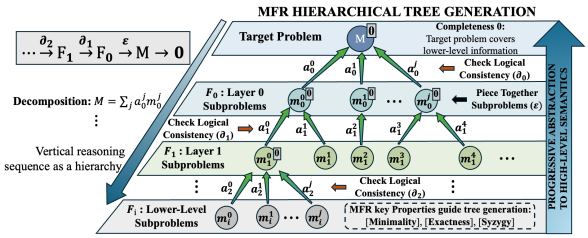

Figure 3: MFR-based hierarchical reasoning in SyRA.

P_5. Since subproblem answers may omit essential context, we provide the root problem as background input for explanation generation:

$$s_i^j = M_{\text{Sc}}\Big(q_i^j, M_{\text{Re}}\big(\text{Con}(q_{\text{root}}, q_i^j), P_4\big), P_5\Big) \quad (8)$$

Where $\text{Con}(\cdot)$ concatenates q_{root} and q_i^j as input. If $s_i^j < \varepsilon$, the node is deemed low-confidence and the RB is activated. KR uses P_6 to extract the information relevant to the node from the layer $i+1$. When there are deeper descendants, it is further filtered to iteratively refine the reasoning.

$$q_{i+t}^k = M_{\text{Ext}}^{(t)}(P_6, q_i^j, s \geq \varepsilon), \quad t = 1, \ldots, T \quad (9)$$

Then, q_{i+t}^k and its reasoning are used to refine node q_i^j. If q_{i+t}^k is a leaf, it is prioritized in backtracking to improve $r_i^{j'}$:

$$r_i^{j'} = M_{\text{Up}}\Big(P_7,\ P\big(F(\{q_{i+t}^k,\ r_{i+t}^k\})\big),\ r_i^j\Big) \quad (10)$$

where M_{Up} is the LLM for backtracking updates, P_7 is the prompt template, $F(\cdot)$ marks leaf nodes as important, and $P(\cdot)$ builds a prompt from fused node information. The final updated node is:

$$n_i^{j'} \leftarrow (\ q_i^j,\ r_i^{j'},\ s_i^j\ ) \quad (11)$$

After completing layer-wise local corrections, all child-node reasoning results n_i^j are incorporated into the prompt with P_8 to guide M_{re} in generating the final reasoning result:

$$r_{\text{root}} = M_{\text{Re}}\big(q_{\text{root}},\ p(\{q_i^j,\ r_i^j,\ P_8\})\big) \quad (12)$$

3.4 MFR Theory and Soundness Justification

Minimal Free Resolution (MFR) is a fundamental algebraic concept for modeling structural dependencies in hierarchical problem decomposition. The MFR of a module M is defined as the following exact sequence (Fieldsteel and Nagel, 2021):

$$\cdots \xrightarrow{\partial_2} F_1 \xrightarrow{\partial_1} F_0 \xrightarrow{\varepsilon} M \to 0 \quad (13)$$

Viewed vertically, the sequence naturally maps to a hierarchical reasoning tree (Figure 3). Arrows ($\to$) represent a progressive abstraction from low-level structures to high-level semantics. Each $F_i = \text{span}\{m_i^j\}$ denotes the space of concurrent subproblems in the reasoning layer i, where m_i^j represents the j-th subproblem. The mappings ∂_i act as dependency projection operators that validate the logical consistency among subproblems. The integration map ε aggregates the minimal subproblems into the final objective M. The terminal 0 indicates the completeness of the information, requiring that higher-level reasoning fully subsumes lower-level derivations. This completeness property also holds recursively: each subproblem m_i^j acts as a local objective whose information must be entirely explained by lower-level structures. To support coherent tree-structured reasoning and cross-layer adaptive error correction in SyRA, we leverage three key concepts from MFR:

Minimality is defined as $\text{Im}(\partial_i) \subseteq \mathfrak{m}F_{i-1}$ (Bruns and Herzog, 1998). In SyRA, this property is realized through the hybrid filtering mechanism, ensuring that F_0 contains strictly the minimal subproblems necessary to resolve the target M, while F_1 captures the structural relationships among the subproblems in F_0.

Exactness is defined as $\text{Im}(\partial_{i+1}) = \text{Ker}(\partial_i)$ (Weibel and Butler, 1996). In SyRA, this property ensures that structural information in layer F_i is fully captured by F_{i+1}, enabling us to define the GCM and IRR metrics to ensure lossless information transfer across reasoning layers.

Syzygy is defined as $\sum_j a_i^j m_i^j = 0$ (Pareschi, 2000). In SyRA, this property encodes cross-layer dependencies among subproblems m_i^j. When reasoning conflicts arise, the KR module uses the coefficients a_i^j to identify contradictions and trigger cross-layer rollback.

In chain-based reasoning, MFR is typically compressed into implicit path-level heuristic constraints, thereby failing to fully realize its structural semantics: characterizing hierarchical subproblem spaces through minimality, ensuring recursive completeness through exactness, and expressing cross-layer dependency propagation through syzygies (Li et al., 2025).

For MFR-based decomposition, Q must first be represented as an algebraic module $M(Q)$ via the algebraic representation $(S, I(Q))$ (Eisenbud, 2013). Here, $S = k[x_1, \ldots, x_n]$ is the symbolic

Model	Method	Math Reasoning					QA	Multitask	Gen.
		GSM8k	SVAMP	MultiArith	AsDiv	AQUA	StrategyQA	BBH	MMLU
Qwen2.5 7B	CoT	0.772	0.804	0.923	0.898	0.606	0.639	0.471	0.553
	DeAR	0.822	0.862	0.955	0.914	0.643	0.673	0.497	0.559
	ToT	0.873	0.936	0.965	0.928	0.555	0.722	0.544	0.562
	SyRA	**0.889**	**0.945**	**0.982**	**0.939**	**0.669**	**0.755**	**0.664**	**0.566**
Qwen2.5 72B	CoT	0.861	0.869	0.973	0.940	0.811	0.734	0.773	0.796
	DeAR	0.896	0.911	0.981	0.949	0.823	0.763	0.791	0.811
	ToT	0.926	0.955	0.988	0.959	0.805	0.749	0.805	0.817
	SyRA	**0.961**	**0.958**	**0.992**	**0.968**	**0.903**	**0.821**	**0.817**	**0.839**
GLM-4 9B	CoT	0.853	0.803	0.904	0.897	0.579	0.682	0.499	0.572
	DeAR	0.871	0.901	0.927	0.920	0.645	0.702	0.532	0.619
	ToT	0.886	0.932	0.931	0.931	0.598	0.729	0.612	0.622
	SyRA	**0.902**	**0.935**	**0.965**	**0.939**	**0.630**	**0.753**	**0.649**	**0.636**
GLM-4 32B	CoT	0.892	0.893	0.936	0.911	0.727	0.693	0.611	0.673
	DeAR	0.921	0.934	0.967	0.930	0.741	0.728	0.688	0.724
	ToT	0.947	0.943	0.983	0.945	0.595	0.742	0.712	0.733
	SyRA	**0.952**	**0.949**	**0.988**	**0.950**	**0.803**	**0.749**	**0.747**	**0.749**
GPT-4o mini	CoT	0.862	0.844	0.953	0.942	0.652	0.721	0.674	0.621
	DeAR	0.926	0.937	0.966	0.947	0.708	0.746	0.733	0.672
	ToT	0.952	0.942	0.984	0.952	0.732	0.767	0.744	0.698
	SyRA	**0.962**	**0.956**	**0.986**	**0.966**	**0.764**	**0.802**	**0.809**	**0.722**

Table 1: Overall results of the framework across eight reasoning datasets.

variable space, where variable x_i corresponds to a key semantic condition (atomic entities) extracted from the problem by a pre-trained T5 model. The ideal $I(Q)$ encodes task constraints derived from logical relations identified by the LLM among these elements. Together, S and $I(Q)$ induce the module structure $M(Q)$, which provides the algebraic foundation for applying MFR.

However, the affine representation induced by $(S, I(Q))$ may not guarantee global minimality, affecting inference consistency (David, 1991). We therefore define the **Homogenized Representation** of Q as $(S^h, I(Q)^h, M(Q)^h)$, where $S^h = k[x_0, x_1, \ldots, x_n]$ is a standard graded polynomial ring with homogenization variable x_0, and $I(Q)^h$ is the homogeneous ideal obtained by homogenizing the constraints in $I(Q)$ (Nguetseng, 2003). Appendix B.1 shows that the homogenization $M(Q)^h$ admits a minimal graded free resolution unique up to isomorphism in S^h, whose uniqueness guides the LLM toward canonical decompositions.

In practice, SyRA realizes the homogenized representation required for MFR-based decomposition by enforcing the minimality and exactness conditions through the KV and KC modules. Appendix B.3 further shows that this construction satisfies the consistency condition, thereby enabling the recursive decomposition of complex problems into structured subproblems.

4 Experiments

4.1 Datasets and Baselines

We use GPT-4o-mini as the primary test model and include Qwen2.5-7B/72B and GLM-4-9B/32B as comparison models (Hurst et al., 2024; Hui et al., 2024; GLM et al., 2024).

We compare SyRA against CoT, DeAR, and ToT as baseline reasoning methods, with detailed settings provided in Appendix E. To assess generalization across diverse tasks, we conduct experiments on five math reasoning datasets (GSM8K, SVAMP, MultiArith, ASDiv, and AQUA), the QA benchmark StrategyQA, the multitask logical reasoning benchmark BBH, and the general knowledge benchmark MMLU (Liu et al., 2023; Chen et al., 2024; Geva et al., 2021; Suzgun et al., 2023; Wang et al., 2024). The Appendix C provides de-

Metric	GSM8K	SVAMP	MultiArith	AsDiv	AQUA	StrategyQA	BBH	MMLU
Branch	1.99	2.12	2.01	1.75	1.56	2.23	2.25	2.13
Depth	2.51	1.65	2.07	1.44	1.62	1.53	1.54	1.49
Length	438.99	330.03	304.13	333.64	430.62	472.55	476.47	468.77

Table 2: Characteristics of T in different datasets.

Method	SVAMP		GSM8K		StrategyQA	
	SC	RA	SC	RA	SC	RA
CoT	0.53	0.63	0.51	0.52	0.42	0.39
DeAR	0.56	0.64	0.55	0.57	0.43	0.41
ToT	0.57	0.66	0.57	0.59	0.45	0.42
SyRA	**0.58**	**0.68**	**0.59**	**0.67**	**0.47**	**0.48**

Table 3: Cross-method automatic evaluation.

Method	SVAMP	GSM8K	StrategyQA
CoT	18	10	13
DeAR	21	16	17
ToT	24	18	20
SyRA	**37**	**56**	**50**

Table 4: Distribution of annotators' selections.

tailed information on the datasets in this paper.

4.2 Applicability Evaluation

Table 1 shows that SyRA performs consistently well across all datasets and different LLMs. Compared with CoT, SyRA's multi-path tree reasoning reduces error accumulation in linear reasoning. Although fixed-branch ToT performs strongly in many settings, SyRA achieves better performance through controllable decomposition and adaptive cross-layer reflection. Although DeAR mitigates ToT's fixed branching, it relies on a preset decomposition example pool, fixed decomposition thresholds, and layer-wise backtracking rules, which may hinder its accuracy and generalization.

To evaluate the impact of LLMs' information extraction on reasoning outcomes, Appendix F.1 presents the results of the KV-KC-KR framework under ground-truth information constraints in different datasets. This reveals that extraction errors in factual tasks (StrategyQA, MMLU) arise mainly from prior knowledge deficits, while failures in math and logic tasks (BBH) arise from limitations in LLM's intrinsic reasoning capabilities. In addition, Appendix F.2 shows that a more advanced information extraction model does not necessarily improve reasoning performance because it may over-identify the core constraints.

4.3 Analysis of the Reasoning Tree

Table 2 summarizes SyRA's reasoning behavior in terms of average branch (Branch), average depth (Depth), and average reasoning length (Length).

For math tasks, base models better capture structured reasoning patterns, improving reasoning quality. For open-domain tasks (StrategyQA, BBH, and MMLU), SyRA's zero-shot decomposition enables LLMs to generate effective subproblems, and longer reasoning chains can still improve performance despite imperfect intermediate steps.

4.4 Reasoning Consistency Evaluation

This study evaluates different reasoning methods using automatic metrics: source consistency (SC) and reasoning alignment (RA), and human assessments(Golovneva et al., 2022). SC measures the consistency between the input problem and the reasoning, while RA measures the alignment between the reasoning and the answer. We focus on SVAMP, GSM8K, and StrategyQA: the first two require structured formal reasoning, while the last involves open-domain. Table 3 shows that SyRA outperforms baselines in all datasets.

For human evaluation, we randomly sampled 100 questions per test instance and had annotators assess the reasoning. Table 4 shows that SyRA

Dataset	Model	Avg Time	ACC	Avg Branch	Avg Depth	Avg Len of R
GSM8K	SyRA (full)	54.98	0.962	1.99	2.51	438.99
	SyRA w/o KV	124.63	0.956	4.11	3.24	450.18
	SyRA w/o KC	43.97	0.951	1.97	1.20	420.12
	SyRA w/o I+G	53.13	0.958	1.79	2.30	431.67
	SyRA w/o KR	39.90	0.897	2.99	2.51	478.50
SVAMP	SyRA (full)	57.33	0.956	2.12	1.65	330.03
	SyRA w/o KV	125.78	0.951	3.91	2.31	349.08
	SyRA w/o KC	49.17	0.948	2.10	1.12	328.14
	SyRA w/o I+G	55.79	0.950	2.08	1.61	330.45
	SyRA w/o KR	40.89	0.942	2.12	1.65	385.12
StrategyQA	SyRA (full)	28.99	0.802	2.43	1.53	472.55
	SyRA w/o KV	33.01	0.799	3.42	1.45	582.53
	SyRA w/o KC	31.07	0.793	2.39	1.73	463.32
	SyRA w/o I+G	28.23	0.796	2.41	1.51	468.98
	SyRA w/o KR	15.11	0.703	2.43	1.53	488.52

Table 5: Model performance on different benchmarks.

achieves higher logical coherence than other methods.

4.5 Trade-off Analysis

Compared with CoT, DeAR, and ToT, SyRA introduces a KV-KC-KR cycle for question decomposition and backtracking, which may affect reasoning efficiency. Appendix D.1 shows the results of each method in terms of average inference time per question (T/Q) and accuracy (ACC).

Although CoT incurs the lowest inference cost, it also yields the lowest accuracy, since later steps often cannot recover from an early deviation in the reasoning path. By contrast, SyRA improves robustness through its multi-stage decomposition and backtracking process, albeit with higher latency due to the KV-KC-KR modules. This trade-off is better suited to complex, accuracy-critical, multi-step reasoning tasks, while CoT may remain preferable for latency-sensitive applications. Appendix D.2 shows that SyRA can identify and revise erroneous intermediate steps through a more human-like reasoning process, while Appendix D.3 reports its average output tokens per question across datasets, providing a reference for practical cost evaluation.

4.6 Ablation Study

We conducted ablation experiments on the filtered data using GPT-4o-mini, with the results shown in Table 5. Removing KV increases subproblem generation (Avg Branch is higher than the full), but does not improve accuracy. This is because redundant subproblems introduce noise that weakens root-level reasoning. Thus, the minimality-based KV module better controls subproblem quality.

Across all datasets, SyRA without KC consistently underperforms full SyRA, suggesting that the exactness-inspired KC module improves the convergence and stability of deep reasoning. Fine-grained results also show that its average depth is lower than full SyRA on structured tasks (GSM8K, SVAMP) but higher on StrategyQA, suggesting that LLMs may not reliably identify the key information needed to decide when to decompose. Moreover, SyRA without KC has a lower average branch than full SyRA, highlighting that IRR and GCM are crucial to preserve information propagation. This raises a key question: how much do these mechanisms affect the model's reasoning?

Under comparable reasoning time, SyRA without I+G consistently underperforms full SyRA, with lower average branch, depth, and reasoning length, indicating that missing information during decomposition weakens reasoning performance. This further underscores the importance of IRR and GCM in KC for reasoning quality.

Although the SyRA without KR removes backtracking and thus reduces runtime, it still underperforms the full in accuracy. Therefore, inspired by syzygy and residual learning, the cross-layer backtracking of RB preserves all key information in the

reasoning tree and selects optimal solution paths (shorter reasoning length) to solve the problem, substantially improving reasoning quality.

5 Conclusion

In this work, we show that tree-structured problem decomposition can be embedded into the hierarchical free-representation space of MFR, alleviating limitations of heuristic-based approaches. Based on this perspective, we propose SyRA, an adaptive reasoning framework inspired by algebraic geometry. Unlike previous works that use mathematics as a metaphor, SyRA operationalizes algebraic theory into algorithmic constraints: minimality for pruning, exactness for consistency, and syzygy for backtracking. Specifically, we develop hybrid filtering for KV to enforce minimality, IRR and GCM for KC to ensure exactness, and RB for KR guided by syzygy. Extensive experiments show that SyRA achieves superior reasoning accuracy and stability across benchmarks, confirming that rigorous algebraic structures effectively guide LLM reasoning.

Acknowledgments

We thank the anonymous reviewers for their insightful and constructive comments. We also gratefully acknowledge support from the National Natural Science Foundation of China Grant No.62277046, the Jiangsu Higher Education Teaching Reform Research Project No.2025JGYB833 and the Open Project Program of Wuhan National Laboratory for Optoelectronics No.2024WNLOKF011.

Limitations

Key Information Extraction: The effective implementation of our method is based on the ability of the LLM to accurately identify the core constraints (the free generating elements) of the problem. We then map the problem to an algebraic module structure and incorporate MFR's minimality, exactness, and syzygy mechanisms to construct the complete SyRA framework. Therefore, future work may require more capable LLMs with stronger key information extraction.

Method Generalizability: This study evaluates SyRA on structured mathematical reasoning tasks, knowledge QA benchmarks, multitask logical reasoning datasets, and general knowledge reasoning tasks to assess the human-like problem-solving ability of LLMs on complex problems. However, for knowledge-intensive reasoning tasks, the analysis of SyRA is limited by the lack of external knowledge. Furthermore, whether this framework can generalize to software development or other end-to-end applications remains an open direction for future work.

Ethics Statement

The dataset used in this study is primarily sourced from publicly available resources. Our data collection process strictly avoids any personally identifiable information, such as user IDs, avatars, or comments, ensuring maximum transparency and accessibility. During our experiments, we provide human participants with a full explanation of data usage and publication; at no point are participants exposed to inappropriate content. We ensured that the whole process adhered to all ethical guidelines and that the time and effort of human participants were used responsibly and transparently, thereby promoting the advancement of research in the field. For all human-subject studies, we strictly adhered to IRB approval. In our experiments, we used human evaluation to assess performance. Two human annotators individually annotated 223 and 177 items, respectively. Participants in this study were compensated $100.

References

M Artale. 1990. A minimal graded free resolution of a non-perfect module. *Journal of Algebra*, 132(1):1–21.

Guangsheng Bao, Hongbo Zhang, Cunxiang Wang, Linyi Yang, and Yue Zhang. 2025. How likely do llms with cot mimic human reasoning? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7831–7850.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 17682–17690.

Winfried Bruns and H Jürgen Herzog. 1998. *Cohen-macaulay rings*. 39. Cambridge university press.

Antonio Capani, Gabriel De Dominicis, Gianfranco Niesi, and Lorenzo Robbiano. 1997. Computing minimal finite free resolutions. *Journal of Pure and Applied Algebra*, 117:105–117.

Kaiyuan Chen, Jin Wang, and Xuejie Zhang. 2024. Mathematical reasoning via multi-step self questioning and answering for small language models. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 81–93. Springer.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Junhang Cheng, Fang Liu, Chengru Wu, and Li Zhang. 2025. Adaptivellm: A framework for selecting optimal cost-efficient llm for code-generation based on cot length. In *Proceedings of the 16th International Conference on Internetware*, pages 461–473.

Pengyu Cheng, Yong Dai, Tianhao Hu, Han Xu, Zhisong Zhang, Lei Han, Nan Du, and Xiaolong Li. 2024. Self-playing adversarial language game enhances llm reasoning. *Advances in Neural Information Processing Systems*, 37:126515–126543.

Cox David. 1991. Ideals, varieties, and algorithms-an introduction to computational algebraic geometry and commutative algebra. *Undergraduate Texts in Mathematics*.

David Eisenbud. 2013. *Commutative algebra: with a view toward algebraic geometry*, volume 150. Springer Science & Business Media.

Nathan Fieldsteel and Uwe Nagel. 2021. Minimal and cellular free resolutions over polynomial oi-algebras. *arXiv preprint arXiv:2105.08603*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Philippe Gimenez, Indranath Sengupta, and Hema Srinivasan. 2010. Minimal free resolutions for certain affine monomial curve. *arXiv preprint arXiv:1011.4247*.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, and 1 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2022. Roscoe: A suite of metrics for scoring step-by-step reasoning. *arXiv preprint arXiv:2212.07919*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning. *Image Recognition*, 7(4):327–336.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Aditya Kalyanpur, Siddharth Patwardhan, BK Boguraev, Adam Lally, and Jennifer Chu-Carroll. 2012. Fact-based question decomposition in deepqa. *IBM Journal of Research and Development*, 56(3.4):13–1.

Roberto La Scala and Michael Stillman. 1998. Strategies for computing minimal free resolutions. *Journal of Symbolic Computation*, 26(4):409–431.

Chenghao Li, Chaoning Zhang, Yi Lu, Jiaquan Zhang, Qigan Sun, Xudong Wang, Jiwei Wei, Guoqing Wang, Yang Yang, and Heng Tao Shen. 2025. Syzygy of thoughts: Improving llm cot with the minimal free resolution. *arXiv preprint arXiv:2504.09566*.

Bingbin Liu, Sebastien Bubeck, Ronen Eldan, Janardhan Kulkarni, Yuanzhi Li, Anh Nguyen, Rachel Ward, and Yi Zhang. 2023. Tinygsm: achieving> 80% on gsm8k with small language models. *arXiv preprint arXiv:2312.09241*.

Zhiyuan Ma, Zhenya Huang, Jiayu Liu, Minmao Wang, Hongke Zhao, and Xin Li. 2025. Automated creation of reusable and diverse toolsets for enhancing llm reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24821–24830.

Kumar Manas, Stefan Zwicklbauer, and Adrian Paschke. 2024. Cot-tl: Low-resource temporal knowledge representation of planning instructions using chain-of-thought reasoning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9636–9643. IEEE.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. *arXiv preprint arXiv:1906.02916*.

Gabriel Nguetseng. 2003. Homogenization structures and applications i. *Zeitschrift für Analysis und ihre Anwendungen*, 22(1):73–108.

Giuseppe Pareschi. 2000. Syzygies of abelian varieties. *Journal of the American Mathematical Society*, 13(3):651–664.

Prasanna Reddy Pulakurthi, Jiamian Wang, Majid Rabbani, Sohail Dianat, Raghuveer Rao, and Zhiqiang Tao. 2025. X-cot: Explainable text-to-video retrieval via llm-based chain-of-thought reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 31172–31183.

Ankit Satpute, Noah Gießing, André Greiner-Petter, Moritz Schubotz, Olaf Teschke, Akiko Aizawa, and Bela Gipp. 2024. Can llms master math? investigating large language models on math stack exchange. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 2316–2320.

Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. 2024. Rl on incorrect synthetic data scales the efficiency of llm math reasoning by eight-fold. *Advances in Neural Information Processing Systems*, 37:43000–43031.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and 1 others. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Charles Weibel and MCR Butler. 1996. An introduction to homological algebra. *Bulletin of the London Mathematical Society*, 28(132):322–323.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198.

Haoyuan Wu, Xueyi Chen, Rui Ming, Jilong Gao, Shoubo Hu, Zhuolun He, and Bei Yu. 2025. Totrl: Unlock llm tree-of-thoughts reasoning potential through puzzles solving. *arXiv preprint arXiv:2505.12717*.

Zhuofeng Wu, Richard He Bai, Aonan Zhang, Jiatao Gu, VG Vinod Vydiswaran, Navdeep Jaitly, and Yizhe Zhang. 2024. Divide-or-conquer? which part should you distill your llm? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2572–2585.

Ziyang Xiao, Dongxiang Zhang, Xiongwei Han, Xiaojin Fu, Wing Yin Yu, Tao Zhong, Sai Wu, Yuan Wang, Jianwei Yin, and Gang Chen. 2024. Enhancing llm reasoning via vision-augmented prompting. *Advances in Neural Information Processing Systems*, 37:28772–28797.

Shangzi Xue, Zhenya Huang, Jiayu Liu, Xin Lin, Yuting Ning, Binbin Jin, Xin Li, and Qi Liu. 2024. Decompose, analyze and rethink: Solving intricate problems with human-like reasoning cycle. *Advances in Neural Information Processing Systems*, 37:357–385.

Xiao-Wen Yang, Xuan-Yi Zhu, Wen-Da Wei, Ding-Chu Zhang, Jie-Jing Shao, Zhi Zhou, Lan-Zhe Guo, and Yu-Feng Li. 2025. Step back to leap forward: Self-backtracking for boosting reasoning of language models. *arXiv preprint arXiv:2502.04404*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Haoyu Zhang, Jingjing Cai, Jianjun Xu, and Ji Wang. 2019. Complex question decomposition for semantic parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4477–4486.

Jinghan Zhang, Xiting Wang, Weijieying Ren, Lu Jiang, Dongjie Wang, and Kunpeng Liu. 2025. Ratt: A thought structure for coherent and correct llm reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26733–26741.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, and 1 others. 2024. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and 1 others. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A KV Module Construction Details

A.1 Training Process

To enable effective decision-making by the subsequent verifier, we identify the core constraints (free generating elements) of the problem statement and train the corresponding constraint discrimination models $F_{\text{gen}}(\cdot)$ using learning-driven heuristic training methods.

Therefore, the original problem Q must first be normalized and broken down into a sequence of sentences:

$$S = Split(Q_{\text{encoded}}) = \{s_1, s_2, \ldots, s_i\}$$

For each sentence s_i, we assign a label $y_i \in \{0, 1\}$. A sentence is labeled as a free generating element if it is essential to solve the problem ($y_i = 1$); otherwise, ($y_i = 0$). Consequently, the task can be formulated as a sentence-level binary classification problem, whose goal is to identify the semantic sentence that critically contributes to

problem solving. To this end, we construct the following dataset:

$$D = \{(Q_i, \{(s_1, y_1), (s_2, y_2), \ldots, (s_k, y_k)\})\}$$

where Q_i denotes the sample of the problem i-th, s_k is sentences, and y_k the corresponding labels. Given the imbalance between positive and negative classes in the datasets (i.e., more effective semantic information than background sentences), we employ a weighted cross-entropy loss to improve the model's ability to recognize the minority class:

$$L_{\text{weighted}} = -\sum_{i=1}^{k}\Big[w_{\text{pos}}\, y_i\, \log p_i + w_{\text{neg}}\,(1-y_i)\,\log(1-p_i)\Big]$$

where w_{pos} and w_{neg} denote the weights of the positive and negative samples. To improve the evaluation stability, we combine the training and validation losses into the overall loss function:

$$L_{\text{total}} = L_{\text{train}} + a \cdot L_{\text{val}}$$

where a is the weighting factor applied to the validation loss to mitigate overfitting.

A.2 Free Generating Elements

Once the model is trained, we can obtain the set of effective free generating elements for each problem:

$$E(Q_i) = \{e_i \mid F_{\text{gen}}(s_i) = 1\}$$

This set captures the core constraints of the problem and the essential inferential basis, and its number is defined as the local generative degree Υ_i:

$$\Upsilon_i = |E(Q_i)|$$

This value measures the diversity of decompositions.

A.3 Question Pruning

The query of Q_i may be a short interrogative sentence, a declarative sentence, or a composite sentence that contains multiple free generating elements.

First, we use a T5 model to decompose the problem Q_i and extract its final original query s_{final}. Next, by combining semantic analysis with rule-based filtering, we prune irrelevant elements from s_{final} to obtain the normalized query s'_{final}:

$$s'_{\text{final}} = Rule(Last(T5\_split(Q_i)))$$

Dataset	Domain	Samples
GSM8K	Math Reasoning	1319
MultiArith	Math Reasoning	600
SVAMP	Math Reasoning	1000
AsDiv	Math Reasoning	2405
AQUA	Math Reasoning	254
StrategyQA	Knowledge QA	2290
BBH	Logical Reasoning	3320
MMLU	General Knowledge	9994

Table 6: Details of all evaluated datasets.

Then, guided by the original and normalized queries and prompts such as "remove at most one sentence," we apply an LLM-assisted pruning algorithm to filter out subproblems that are semantically redundant with the original query:

$$Q_i^* = GP(Fil(q_i, s_{\text{f}}, s'_{\text{f}}, P), \text{constraint: } del \leq 1)$$

where q_i is the candidate subproblems generated by the LLM. $\text{Fil}(q_i, s_{\text{f}}, s'_{\text{f}}, P)$ removes the subproblem candidate that is semantically most similar to s_{f} or s'_{f} under the guidance of the prompt P. GP indicates performing additional pruning operations under the given constraints.

B Mathematical Proofs

B.1 Proof of Theorem 1 (Global Existence)

Theorem 1 (Global Existence of Minimal Graded Free Resolution)**.** (Artale, 1990) Let $M(Q)^h$ denote the homogenized module for problem Q. Then $M(Q)^h$ admits a **unique Minimal Graded Free Resolution** over the global ring S^h. That is, there exists an exact sequence of graded free modules:

$$0 \to F_p \xrightarrow{\partial_p} \ldots \xrightarrow{\partial_1} F_0 \xrightarrow{\varepsilon} M(Q)^h \to 0 \quad (14)$$

where ∂_i is degree-preserving, and the image of ∂_i is contained in $\mathfrak{m}F_{i-1}$, with $\mathfrak{m} = (x_0, \ldots, x_n)$ being the irrelevant maximal ideal of S^h.

The proof proceeds by inductively constructing a resolution based on the structural properties of standard graded rings and the Graded Nakayama's Lemma.The specific steps are detailed as follows:

1. Algebraic Setting. Let $S^h = k[x_0, \ldots, x_n]$ be the standard graded polynomial ring. Note that S^h is a positively graded ring where $(S^h)_0 = k$ is a field. Let $\mathfrak{m} = \bigoplus_{d>0}(S^h)_d = (x_0, \ldots, x_n)$ denote the unique irrelevant maximal ideal. Since

$I(Q)^h$ is a homogeneous ideal, the quotient $M(Q)^h = S^h/I(Q)^h$ is a finitely generated graded S^h-module.

2. Base Case: Construction of F_0. Consider the quotient vector space $V_0 = M(Q)^h/\mathfrak{m}M(Q)^h$. Since $M(Q)^h$ is finitely generated, V_0 is a finite-dimensional graded vector space over k. Let $\{\bar{e}_1, \ldots, \bar{e}_{\beta_0}\}$ be a homogeneous basis for V_0, with degrees $d_j = \deg(\bar{e}_j)$.

By the **Graded Nakayama's Lemma**, any set of homogeneous lifts $\{e_1, \ldots, e_{\beta_0}\} \subset M(Q)^h$ forms a minimal generating set for $M(Q)^h$. Accordingly, we define the first free module as:

$$F_0 = \bigoplus_{j=1}^{\beta_0} S^h(-d_j)$$

We construct the surjective homomorphism $\varepsilon : F_0 \to M(Q)^h$ by mapping the basis elements of F_0 to the generators $\{e_j\}$.

3. Inductive Step: Syzygies. Let $K_0 = \ker(\varepsilon)$. Since S^h is a Noetherian ring, K_0 is a finitely generated graded submodule of F_0. A crucial consequence of choosing a *minimal* generating set is that $K_0 \subseteq \mathfrak{m}F_0$. This inclusion represents the algebraic condition for minimality: it ensures that the relation matrix contains no units (constants), preventing trivial redundancies.

We proceed by induction. Assume F_i and the map $\partial_i : F_i \to F_{i-1}$ have been constructed such that $K_i = \ker(\partial_i) \subseteq \mathfrak{m}F_i$. We construct F_{i+1} by applying the base case procedure to K_i. Specifically, we find a minimal generating set for K_i, define F_{i+1} accordingly, and construct the map $\partial_{i+1} : F_{i+1} \to F_i$ covering these generators. By construction, the image $\mathrm{Im}(\partial_{i+1}) = K_i$ is contained in $\mathfrak{m}F_i$.

4. Termination. To guarantee this process does not continue indefinitely, we invoke the **Hilbert Syzygy Theorem**. Since $S^h = k[x_0, \ldots, x_n]$ is a polynomial ring in $n+1$ variables over a field, it has finite global dimension. Specifically, the projective dimension of the module is bounded by the number of variables:

$$\mathrm{pd}_{S^h}(M(Q)^h) \leq n + 1$$

This inequality implies that for some $p \leq n + 1$, the kernel K_{p-1} must be a free module. We then set $F_p = K_{p-1}$ and ∂_p as the inclusion map, terminating the sequence.

5. Uniqueness. The minimality of $\mathrm{Im}(\partial_i) \subseteq \mathfrak{m}F_{i-1}$ implies that the rank of each free module F_i is an intrinsic invariant. Specifically, tensoring the minimal resolution with the residue field $k = S^h/\mathfrak{m}$ yields a complex with zero differentials, isomorphic to the Tor groups:

$$F_i \otimes_{S^h} k \cong \mathrm{Tor}_i^{S^h}(M(Q)^h, k)$$

The ranks are thus determined by graded Betti numbers $\beta_{i,j} = \dim_k \mathrm{Tor}_i^{S^h}(M(Q)^h, k)_j$, which are independent of the choice of basis. Consequently, the constructed minimal graded free resolution is unique up to an isomorphism of chain complexes.

B.2 Operationalizing Homogenization

KV as Homogenization Operator: KV uses a hybrid filtering mechanism based on free generating elements that operationalizes the unique minimal generating set implied by projective homogenization. This process serves as an analogue of the homogenization variable x_0 in $S^h = k[x_0, \ldots, x_n]$, enforcing redundancy elimination and independence to ensure a minimal decomposition.

KC as Scale-Invariance Enforcer: Homogenization establishes a scale-invariant structure in which algebraic properties persist across depths. In SyRA, KC operationalizes this invariance by basing all inter-layer decisions on Inter-layer Retention Rate (IRR) and Global Coverage Mechanism (GCM). This uniform application ensures consistent Exactness in information propagation, regardless of the level of problem abstraction.

B.3 Proof of Corollary 1 (Inheritance of Structure)

Corollary 1 (Inheritance of Minimal Resolution)**.** Let decomposition map $\Delta : Q \to \{Q'_1, \ldots, Q'_k\}$ be algebraically consistent, such that each subproblem Q'_j corresponds to a generator of the **first syzygy module** (the algebraic kernel) of the parent module $M(Q)^h$(Capani et al., 1997).

By the Noetherian property of S^h, each associated $M(Q'_j)^h$ is a finitely generated graded module. Consequently, each subproblem admits a **unique Minimal Graded Free Resolution**, ensuring that structural optimality is preserved recursively at every depth of the reasoning tree(La Scala and Stillman, 1998). The specific steps are as follows:

The validity of the recursive step is grounded in the stability of the Noetherian property under submodule construction and the sequential nature of free resolutions. In the SyRA framework,

Dataset	Metric	CoT	DeAR	ToT	SyRA
SVAMP	T/Q	39.77	58.91	67.25	57.33
	ACC	0.844	0.937	0.942	**0.956**
GSM8K	T/Q	38.16	57.67	62.19	54.98
	ACC	0.862	0.926	0.952	**0.962**
StrategyQA	T/Q	21.11	29.33	30.18	28.99
	ACC	0.721	0.746	0.767	**0.802**

Table 7: Accuracy (ACC) and time per question (T/Q) across methods.

	GSM8K	SVAMP	MultiArith	ASDiv	AQUA	StrategyQA	BBH	MMLU
Output	977.41	569.31	635.49	564.97	557.86	675.29	621.67	633.18

Table 8: Average number of output tokens per question across different datasets.

the heuristic decomposition of a problem Q into subproblems $\{Q'_j\}$ corresponds algebraically to identifying the internal dependencies among the generators of the reasoning space. Formally, let $F_0 \xrightarrow{\varepsilon} M(Q)^h \to 0$ denote the initial step of the minimal resolution established in Theorem 1. The set of subproblems $\{Q'_j\}$ naturally generates the kernel of this map, denoted as $K_0 = \ker(\varepsilon) \subseteq F_0$, which is known in commutative algebra as the first syzygy module $\mathrm{Syz}_1(M(Q)^h)$. Thus, solving the subproblems is equivalent to finding a minimal generating set for K_0.

To ensure that this decomposition is computationally feasible and mathematically well-defined, we appeal to the fundamental properties of the underlying ring structure. Since the homogenized polynomial ring $S^h = k[x_0, \ldots, x_n]$ is Noetherian by Hilbert's Basis Theorem, and F_0 is constructed as a finitely generated free module, it follows that any submodule of F_0 must also be finitely generated. Consequently, the kernel K_0 is a finitely generated graded S^h-module. This algebraic fact provides the critical guarantee for the SyRA framework: the number of subproblems (syzygies) required to resolve the parent node is necessarily finite, thereby preventing any infinite regression of dependencies during the reasoning process.

Having established that the subproblem module K_0 is a finitely generated graded module over a Noetherian ring, it satisfies all the preconditions of Theorem 1. Therefore, K_0 itself admits a unique (up to isomorphism) Minimal Graded Free Resolution. Moreover, this resolution is intrinsically linked to the global structure of the parent problem; specifically, the minimal resolution of K_0 corresponds precisely to the truncated complex $0 \to F_p \to \cdots \to F_1 \to K_0 \to 0$, where the free modules are identical to those of $M(Q)^h$ with indices shifted by one. This confirms that the "child" nodes in the SyRA reasoning tree inherit the structural optimality of the "parent" node, providing rigorous mathematical justification for the recursive application of the KV and KC modules.

C Data Description

Table 6 summarizes the datasets used in this study. These datasets have two categories: five structured math datasets, where AQUA contains multiple-choice questions and the others involve direct arithmetic problems, and two open-domain knowledge reasoning datasets, namely StrategyQA for question answering and BBH for multi-step reasoning. All datasets are accessible via HuggingFace.

D Analysis of Reasoning Behavior

D.1 Efficiency and Accuracy Analysis

Table 7 shows the results of each method in terms of average inference time per question (T/Q) and accuracy (ACC). For the compared baselines, we use detailed step-by-step reasoning to ensure their best possible accuracy.

D.2 Analysis of Reasoning Errors

The original problem is stated as follows: Carlos is planting a lemon tree. The tree costs $90 to plant. Each year, it yields 7 lemons, each of which can be

Module	Parameter	Value / Description
Knowledge Verifier (KV)	w_{pos}	1.0 (Positive class weight in Eq. A.1)
	w_{neg}	2.5 (Negative class weight in Eq. A.1)
	a	0.5 (Validation loss weight in Eq. A.1)
	del_{max}	1 (Max sentence removal constraint in A.3)
	Base Model	*Text-To-Text Transfer Transformer small (T5-small)*
Knowledge Controller (KC)	Υ_{min}	2 (Min decomposition threshold)
	ϵ_{IRR}	1e-5 (Stabilizer for IRR calculation)
	R_{trigger}	1.0 (Threshold to trigger GCM if IRR < 1.0)
	Max Depth	*6 (Global hard limit for tree depth)*
	Max Branch	*5 (Max subproblems per node)*
Knowledge Reflector (KR)	ϵ_{score}	0.8 (Consistency score threshold for RB)
LLM Config	Temperature	0.7 (Reasoning), 0.0 (Scoring/Extraction)
	Top-p	0.95
	Max Tokens	1024
	Model Version	gpt-4o-mini-2024-07-18 Qwen2.5-72B/7B-Instruct THUDM/GLM-4-32B/9B-0414

Table 9: Hyperparameter settings for SyRA modules and LLM backbone.

	GSM8K	SVAMP	MultiArith	ASDiv	AQUA	StrategyQA	BBH	MMLU
ACC	25%	20%	10%	15%	25%	65%	30%	50%

Table 10: Accuracy of SyRA on error cases given correct element extraction.

sold for $1.5. Watering and feeding the tree cost $3 per year. How many years will it take before the tree starts generating profit?

Figure 4 shows that, due to insufficient prior knowledge, the model confuses break-even with profit. After injecting relevant knowledge, it correctly identifies the 13th year as the point at which profit begins and arrives at the correct answer.

D.3 Output Tokens Per Question

Table 8 shows the average number of output tokens per question generated by SyRA across datasets, providing a reference for practical cost evaluation.

E Hyperparameter Settings

Table 9 summarizes the key hyperparameters used in SyRA across all experiments. We explicitly list the thresholds used for the Knowledge Controller (KC) and the Knowledge Reflector (KR), as well as the training parameters for the Knowledge Verifier (KV). In addition, for CoT, we adopt a detailed 5-step prompting strategy; for DeAR and ToT, we set the branching factor to 4 and the maximum depth to 3.

F Information Extraction Analysis

F.1 Impact of Extraction Errors

Given the limited number of error cases, we randomly selected 20 failure cases from each dataset and re-evaluated them under ground-truth constraints within the KV-KC-KR framework to assess the impact of extraction errors. Results are shown in Table 10.

Method	GSM8K	SVAMP	MultiArith	ASDiv	AQUA	StrategyQA	BBH	MMLU
SyRA-T5	0.962	0.956	0.986	0.966	0.764	0.802	0.809	0.722
SyRA-gpt-4o-mini	0.958	0.953	0.984	0.961	0.759	0.799	0.804	0.719

Table 11: Comparison of SyRA with Different Extraction Models.

Datasets	SyRA	Time	ACC	Branch	Depth	Length
GSM8K	T5	54.98	0.962	1.99	2.51	438.99
	gpt-4o-mini	55.61	0.958	2.01	2.52	440.13
SVAMP	T5	57.33	0.956	2.12	1.65	330.03
	gpt-4o-mini	57.96	0.953	2.13	1.67	331.78
StrategyQA	T5	28.99	0.802	2.43	1.53	472.55
	gpt-4o-mini	29.16	0.799	2.46	1.57	477.89

Table 12: Performance and Reasoning Behavior of SyRA with Different Extraction Models.

```
Q0: Carlos is ... How many years ... he starts earning money?
|-- Q0.1 (From Q0; μ=2 -> Continue Decomposition)
|   How much profit ... from the lemon tree each year?
|   -> Answer: 7.5
|   |-- Q0.1.1 (From Q0.1; μ=1 -> Atomic, Terminate)
|   |   How much money ... from selling lemons?
|   |   -> Answer: 10.5
|   `-- Q0.1.2 (From Q0.1; μ=1 -> Atomic, Terminate)
|       How much ... to water and feed the tree?
|       -> Answer: 3
`-- Q0.2 (From Q0; μ=2 -> Continue Decomposition)
    How many years does it take to recover the planting cost?
    -> Answer: 12
    |-- Q0.2.1 (From Q0.2; μ=1 -> Atomic, Terminate)
    |   How much does it cost to plant the lemon tree?
    |   -> Answer: 90
    `-- Q0.2.2 (From Q0.2; μ=1 -> Atomic, Terminate)
        What is the yearly profit from the lemon tree?
        -> Answer: 7.5
```

Figure 4: Errors from insufficient prior knowledge

F.2 Effect of Extraction Models

Table 11 shows the modularity of the SyRA framework, in which T5 can be directly replaced by more powerful models (gpt-4o-mini) that utilize prompts to extract the constraints of the core problem.

Table 12 compares representative samples in runtime (Time), accuracy (ACC), average branch (Branch), average depth (Depth), and average length of reasoning (Lenth). Although SyRA-gpt-4o-mini records higher metrics in Time, Branch, Depth, and Length, it yields lower ACC than SyRA-T5. This suggests that SyRA-gpt-4o-mini is more susceptible to generating redundant or erroneous constraints, which compromises the final reasoning performance.