# Towards Formalizing Socratic Questions for Explainable Socratic Question Generation

Surawat Pothong[1] Paul Reisert[2] Naoya Inoue[1,3] Machi Shimmei[4] Wenzhi Wang[1,3]
Shoichi Naito[1,3,5] Jungmin Choi[3] Kentaro Inui[6,4,3]
[1]JAIST [2]Beyond Reason [3]RIKEN
[4]Tohoku University [5]Ricoh Company, Ltd. [6]MBZUAI
{spothong, naoya-i}@jaist.ac.jp, beyond.reason.sp@gmail.com, machi.shimmei.e6@tohoku.ac.jp
{wang.wenzhi.r7, naito.shoichi.t1,}@dc.tohoku.ac.jp, jungmin.choi@riken.jp, kentaro.inui@mbzuai.ac.ae

## Abstract

Socratic questioning (SQ) is an effective strategy for fostering critical thinking. One of the key requirements for using SQs in educational settings is maintaining transparency and logical alignment with the content. For generating pedagogically appropriate SQs, we explore a logic-based template approach by first leveraging argumentative components. We conduct an annotation on top of argument-SQ pairs and achieve moderate inter-annotator agreement (Cohen's Kappa: 0.49) and 84% for annotating SQ components. We analyze areas of disagreement, offering insights for curating a template set. This work lays a foundation for advancing template-based Natural Language Question Generation methods and improving model transparency.

## 1 Introduction

Socratic questioning is a structured method of inquiry used to explore complex ideas, uncover truths, analyze concepts, and reveal assumptions [1]. Unlike regular questioning, it considers key principles, theories, and problems in a systematic way[2, 3] .

SQ is a tool for fostering critical thinking and addressing cognitive biases, but its adoption is limited by the difficulty instructors face in manually crafting context-specific questions for each scenario [1, 2]. This hinders scalability and effectiveness in educational settings. To address this, we explore a template-based approach that automates SQ generation by leveraging argumentative components to clarify the connections between questions and content.

Recent advances in Natural Language Understanding have focused on automating SQ generation as an answer-unaware task to foster critical thinking and self-reflection [4]. Studies leveraging advanced language models such as GPT-2, T5, and ProphetNet [5, 6, 7] have introduced datasets and models for SQ generation. While these approaches enable tasks like cognitive reframing [8], they often produce repetitive or irrelevant outputs, leading to confusion and reducing their effectiveness [9, 10]. Moreover, end-to-end models lack transparency, as their black-box mechanisms fail to clarify how generated questions align with content [11, 12, 13].

To address this, we aim to incorporate argumentation theory to enhance transparency, inspired by Walton's argumentation schemes [14]. Compared to Walton' s critical questions, which effectively evaluate an argument' s logical consistency and evidence, SQ offers distinct advantages by fostering deeper exploration of ideas and a richer understanding of their conceptual foundations [2]. Additionally, recent work in in NLU has demonstrated the effectiveness of templates in capturing reasoning patterns. Logical templates and slot-filling techniques have been applied successfully to identify valid reasoning [15], detect fallacies [16], and model counter-argument logic [17]. These approaches highlight the value of templates in providing structured and interpretable representations of complex logical structures, offering a robust framework for deeper reasoning and analysis.

Building on existing work, we explore a logic-based template and slot-filling approach to enhance the transparency and explainability of SQ generation. Our method explicitly links content to generated questions, capturing the logical structure of arguments and their inferences to produce more transparent and meaningful outputs. This

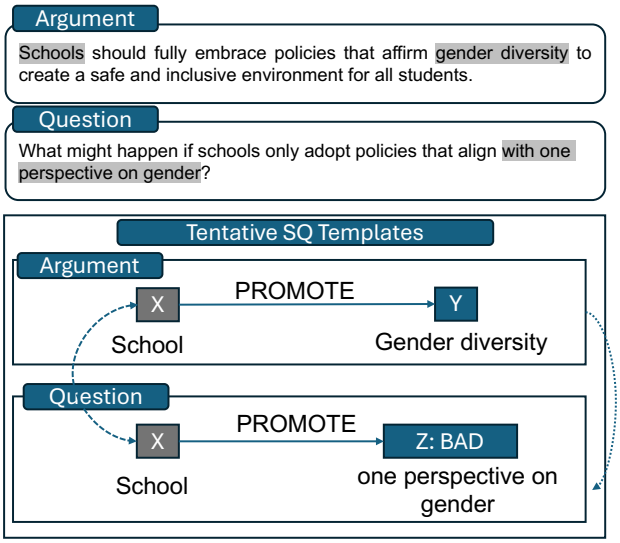


Figure 1: Overview of SQ predicates designed to probe biases or flaws in understanding. The figure illustrates the logical connections between arguments and questions using templates and slot-filling techniques. Slot fillers $X$, $Y$, and $Z$ represent key elements, while predicates explicate the logical relationships among them. The figure specifically demonstrates the "Probing Implication and Consequences" SQ type, which aims to probe the impacts or implications of a thought.

approach addresses the limitations of end-to-end models while providing a systematic framework to help learners develop critical thinking skills.

To guide our research, we pose the following question: To what extent can we create a repository of predicates that capture the structural patterns of SQ to benefit interpretable question generation tasks? To address this, we build upon the dataset introduced in [6] by developing argumentative components that encapsulate the structural logic of SQ. Our framework reformulates the task as a combination of template selection and slot filling, explicitly representing the logical connections between arguments and questions. Figure 1 illustrates our proposed approach, supported by predicate annotations at both argument and question levels. We report an inter-annotator agreement (Cohen's Kappa score) of 0.49 and a predicate coverage of 84% across 50 instances from the SoQG test set. Additionally, we analyze disagreements and distribution patterns, highlighting the potential for explainable template formulation. This work lays the foundation for future large-scale annotations aimed at further enhancing transparency in SQ generation models.

# 2 Towards Formulating Templates

## 2.1 Design Principles for Predicates

**Explaining Underlying Connections and Question Intentions by Coverage** We aim to formulate templates by first developing predicate-level representations, which serve as the foundation for template construction. To address the opacity of end-to-end models, our primary goal is to explicitly establish the relationship between arguments and the questions generated. These predicates are designed to be interpretable by humans, thereby facilitating an understanding of model behavior. We evaluate the interpretability of our approach through the predicates' coverage score, as assessed by human judgment.

**Ease of Annotation** We design straightforward and intuitive predicates, enabling annotators to apply them consistently while achieving an adequate Inter-Annotator Agreement (IAA) score, measured using Cohen's Kappa, consistent with [17, 18].

**Alignment with Socratic Objectives** The predicates are carefully designed to align with the core objectives of SQ. They are intended to promote thoughtful exploration, foster critical analysis, and encourage deeper understanding, thereby reinforcing the pedagogical goals of Socratic dialogue.

## 2.2 Socratic Predicate Inventory

We create the predicate inventory using the "M_Turk_Test set" from Ang et al., preprocessing it with a 300-character limit and filtering out irrelevant questions to ensure contextual relevance [8, 6]. The predicates explicitly represent logical relationships between questions and content, improving interpretability and transparency. Our focus includes five types of SQ: (1) Alternative Viewpoint, (2) Probing Assumption, (3) Probing Implication and Consequences, (4) Probing Reason and Evidence, and (5) Clarification.

We formulate the task of SQ predicate instantiation as follows: Given an original argument $A$ and a Socratic question $SQ$, we first identify a relation $R$ in $A$, composed of slot-fillers $S_1$ and $S_2$, where one of the slot-fillers is related to an important keyword $Z$ in $SQ$. Figure 2 illustrates an example of the inventory, where predicates are derived from the Argument from Consequences and the Argument from

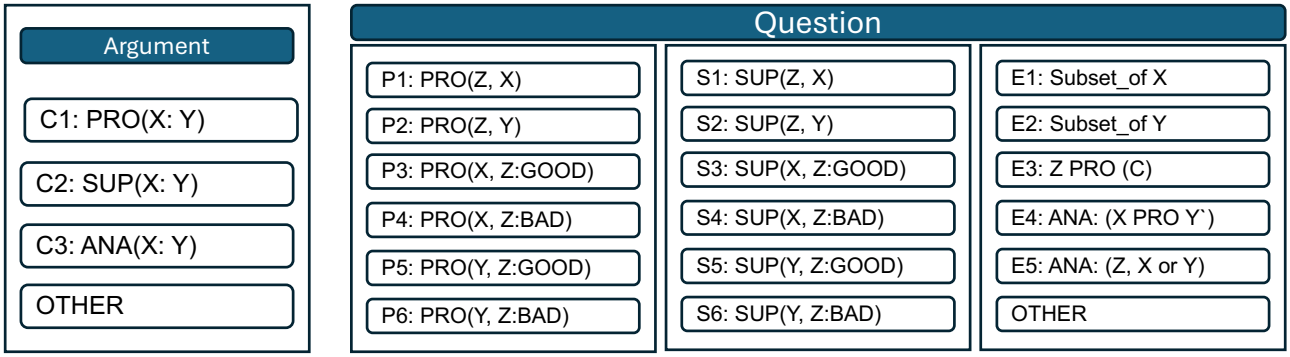


Figure 2: Inventory of Proposed Predicates: The argument level consists of four predicates, while the question level includes 16 predicates. Annotators select one predicate from each level when annotating the SoQG dataset. The argument level is represented by the acronym *C*, and the question level is categorized into three groups: *P* for Promote, *S* for Suppress, and *E* for Extra. The Extra group includes specific categories such as Subset and ANA (Analogical), providing detailed distinctions for comprehensive annotation.

Analogy [14], as well as counter-argument templates [17]. The content and question predicates are represented by four types of relations: **PROMOTE**, **SUPPRESS**, **ANALOGOUS**, and **OTHER**. This results in four types of initial argument predicates and sixteen question predicates.

Two variables, *X* and *Y*, are used to represent the slot fillers in the initial argument. For the sixteen question predicates, we introduce *Z*, inspired by [17], to represent an additional slot filler that captures the question logic in connection to the argument. Additionally, in the question predicates, we include sentiment labels such as **GOOD** and **BAD** to make the predicates more comprehensive by reflecting sentiment nuances.

### 2.3 Annotation Guideline

To construct annotation guidelines, one annotator first independently created logic predicates for arguments and SQs. Inspired by [17], the "Alternative Viewpoint" type was focused on first, aligning it with existing counter-argument templates. A second annotator reviewed and discussed results to evaluate agreement and refine insights. Expanding to other SQ types, disagreements were resolved and multiple annotations were conducted to establish the gold standard.

## 3 Pilot Annotation Study

In this section, we conduct a trial annotation study to evaluate the feasibility of our proposed predicates on top of an existing dataset of SQs. We assess our predicates in terms of coverage and **I**nter-**A**nnotator **A**greement(IAA) on a test set from the SoQG dataset. Finally, we present the results and analysis to gain insights from the annotation study.

### 3.1 Annotating the development set

For our annotation study, we utilize an existing dataset of SQ provided by Ang et al. [6] referred to as SoGQ dataset. SoGQ dataset consists of 110 instances, each comprising an argument and a Socratic question extracted from Reddit's Change My View subreddit[1)], along with annotations for the Socratic question type.

After establishing annotation guidelines, we sampled 50 instances (10 per SQ type) from the SoGQ dataset, filtering irrelevant questions. Using the test set as the development set, two annotators evaluated the templates for coverage and ease of annotation.

### 3.2 Results and Analysis

Table 3 shows the distribution of argument-level predicate annotations between both annotators. Both labeled 25 instances with the **PROMOTE** relation and agreed on 12 instances for **SUPPRESS**, making these the most frequent predicates.The **ANALOGOUS** relation showed lower agreement, reflecting the rarity of analogy-based arguments. The **OTHERS** relation, appearing in 9 instances, highlights the need to consider additional components.

We report coverage and inter-annotator agreement (IAA) using Cohen's Kappa [19] on 50 instances annotated by two annotators. Table 1 shows significant agreement, with

1) https://www.reddit.com/r/changemyview/

Table 1: Coverage results show the instances where the proposed predicates can be instantiated by both annotators out of the 50 instances.

| Annotator | Annotated Instance | Coverage |
|---|---|---|
| Annotator 1 | 42 | 0.84 |
| Annotator 2 | 39 | 0.78 |

coverage scores of 84% and 78%, demonstrating the robustness of the predicates for the SoQG dataset. Instances not covered by the three predicates are labeled as "Other." Therefore, a significant challenge in SQ predicate annotation arises from the complexity of arguments, where multiple logical connections may exist within a single argument. This often leads to ambiguity, as arguments can be mapped to multiple patterns during annotation.

Additionally, Table 2 presents the IAA measured using Cohen's Kappa for 50 instances. Based on [19], we obtained a moderate agreement score, with an average of 0.4917 for the combined argument and question predicates. To further evaluate the performance, we conducted an ablation test at the predicate level by removing sentiment from the predicates to assess its impact. This resulted in a slight increase in the average score to 0.5017, indicating that sentiment has a relatively minor effect on predicate annotations. Furthermore, we observed that some question-level predicates could be combined, such as merging *S1* and *S2* into *S1* and *S3*, *S4*, *S5*, *S6* into *S4* (similarly for *P*). After combining these semantically similar predicates, the score significantly increased to 0.5604, highlighting the effectiveness of reducing redundancy in the predicate set.

## 3.3 Discussion

**Disagreement Discussion** Upon investigating the sources of annotator disagreement, we identified three primary types of errors contributing to the discrepancies: swapped slot fillers, absence of sentiment in the content, and lack of content with no suitable predicate available. For instance, swapped slot fillers occur when an annotator assigns the same slot fillers for *X* and *Y* but inadvertently swaps them. This results in differences when selecting predicates for content and questions, leading to inconsistent relationships in the predicates.

**Implicit Elements and Question Intentions** Through the logical predicates, annotators observed instances with implicit elements during the annotation process, where the question attempts to probe underlying arguments made by the argument author, as illustrated in Figure 1. Some SQ explanations stem from the SQ author's beliefs or assumptions (e.g., perceiving a discrepancy between their interpretation and the Content). In other cases, the SQ author aims to explore the implicit logic embedded within the original argument provided by the Content author.

Table 2: Table showing the IAA with Cohen's Kappa scores. We present the raw scores, scores after removing sentiment from the predicates, and scores after combining semantically similar predicates.

| Category | General | w/o sentiment | w/o sentiment + combined predicates |
|---|---|---|---|
| Arguments | 0.5486 | 0.5486 | 0.5486 |
| Questions | 0.4348 | 0.4549 | 0.5723 |
| Average (All) | 0.4917 | 0.5017 | 0.5604 |

**Potential Templates** After annotating the predicates and calculating the IAA, we identified patterns across argument-level and question-level predicates, forming a basis for constructing SoQG templates that explicate logical connections between arguments and SQ. Tables 4, 5, and 6 present the distribution of annotated question-level predicates relative to argument-level predicates. The analysis shows that the **PROMOTE** relation frequently associates with *P2: PRO(Z, Y)* and *E5: ANA(Z, X or Y)*, while the **SUPPRESS** relation aligns with *P2: PRO(Z, Y)* and *S2: SUP(Z, Y)*. The **ANALOGOUS** relation primarily links to *E5: ANA(Z, X or Y)*, capturing analogies or shared characteristics. These findings highlight the potential to refine templates for SQ generation, ensuring a clear and logical connection between argument-level and question-level predicates.

# 4 Conclusion

Towards formalizing SQs for explainable generation, we explored template techniques to clarify logical connections between arguments and questions. A pilot annotation of 50 argument-SQ pairs achieved moderate IAA (0.50 Cohen's Kappa) with significant coverage. Our analysis identified useful components, which will guide future template curation and large-scale annotation.

## Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 22H00524 and the Nakajima Foundation. We thanks Apisornpanich Latita, Tungrut Wissanu ,and Shall Jaiden for their generous support.

## References

[1] Richard Paul and AJA Binker. Socratic questioning. **Critical thinking: What every person needs to survive in a rapidly changing world**, pp. 269–298, 1990.

[2] Richard Paul and Linda Elder. **The thinker's guide to Socratic questioning**. Rowman & Littlefield, 2019.

[3] Richard Paul and Linda Elder. Critical thinking: The art of socratic questioning. **Journal of developmental education**, Vol. 31, No. 1, p. 36, 2007.

[4] Frederick F Schauer. **Thinking like a lawyer: a new introduction to legal reasoning**. Harvard University Press, 2009.

[5] Erfan Al-Hossami, Razvan Bunescu, Ryan Teehan, Laurel Powell, Khyati Mahajan, and Mohsen Dorodchi. Socratic questioning of novice debuggers: A benchmark dataset and preliminary evaluations. In Ekaterina Kochmar, Jill Burstein, Andrea Horbach, Ronja Laarmann-Quante, Nitin Madnani, Anaïs Tack, Victoria Yaneva, Zheng Yuan, and Torsten Zesch, editors, **Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)**, pp. 709–726, Toronto, Canada, July 2023. Association for Computational Linguistics.

[6] Beng Heng Ang, Sujatha Das Gollapalli, and See-Kiong Ng. Socratic question generation: A novel dataset, models, and evaluation. In Andreas Vlachos and Isabelle Augenstein, editors, **Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics**, pp. 147–165, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[7] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. **Journal of machine learning research**, Vol. 21, No. 140, pp. 1–67, 2020.

[8] Anmol Goel, Nico Daheim, and Iryna Gurevych. Socratic reasoning improves positive text rewriting, 2024.

[9] Wenting Zhao, Ge Gao, Claire Cardie, and Alexander M Rush. I could've asked that: Reformulating unanswerable questions. **arXiv preprint arXiv:2407.17469**, 2024.

[10] Xuan Long Do, Bowei Zou, Shafiq Joty, Tran Tai, Liangming Pan, Nancy Chen, and Ai Ti Aw. Modeling what-to-ask and how-to-ask for answer-unaware conversational question generation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 10785–10803, Toronto, Canada, July 2023. Association for Computational Linguistics.

[11] Shasha Guo, Lizi Liao, Cuiping Li, and Tat-Seng Chua. A survey on neural question generation: Methods, applications, and prospects, 2024.

[12] Artidoro Pagnoni, Alexander R Fabbri, Wojciech Kryściński, and Chien-Sheng Wu. Socratic pretraining: Question-driven pretraining for controllable summarization. **arXiv preprint arXiv:2212.10449**, 2022.

[13] Jing Gu, Mostafa Mirshekari, Zhou Yu, and Aaron Sisto. ChainCQG: Flow-aware conversational question generation. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume**, pp. 2061–2070, Online, April 2021. Association for Computational Linguistics.

[14] DN Walton. **Argumentation schemes**. Cambridge University Press, 2008.

[15] Paul Reisert, Naoya Inoue, Tatsuki Kuribayashi, and Kentaro Inui. Feasible annotation scheme for capturing policy argument reasoning using argument templates. In Noam Slonim and Ranit Aharonov, editors, **Proceedings of the 5th Workshop on Argument Mining**, pp. 79–89, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[16] Irfan Robbani, Paul Reisert, Naoya Inoue, Surawat Pothong, Camélia Guerraoui, Wenzhi Wang, Shoichi Naito, Jungmin Choi, and Kentaro Inui. Flee the flaw: Annotating the underlying logic of fallacious arguments through templates and slot-filling. **arXiv preprint arXiv:2406.12402**, 2024.

[17] Shoichi Naito, Wenzhi Wang, Paul Reisert, Naoya Inoue, Camélia Guerraoui, Kenshi Yamaguchi, Jungmin Choi, Irfan Robbani, Surawat Pothong, and Kentaro Inui. Designing logic pattern templates for counter-argument logical structure analysis. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, **Findings of the Association for Computational Linguistics: EMNLP 2024**, pp. 11313–11331, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[18] Mary L McHugh. Interrater reliability: the kappa statistic. **Biochemia medica**, Vol. 22, No. 3, pp. 276–282, 2012.

[19] Matthijs J Warrens. Five ways to look at cohen's kappa. **Journal of Psychology & Psychotherapy**, Vol. 5, , 2015.

# A Appendix

## A.1 Potential Templates Distribution

Table 3: The distribution of annotated content level predicates among two annotators.

| Argument | Annotator 1 | Annotator 2 |
|---|---|---|
| C1: PRO(X: Y) | 25 | 25 |
| C2: SUP(X: Y) | 12 | 17 |
| C3: ANA(X: Y) | 4 | 3 |
| OTHER | 9 | 5 |

Table 4: Frequency distribution of question predicates associated with content predicate C1: PRO(X,Y)

| Question | Annotator 1 | Annotator 2 |
|---|---|---|
| P1: PRO(Z, X) | 0 | 3 |
| P2: PRO(Z, Y) | **7** | **3** |
| P3: PRO(X, Z:GOOD) | 0 | 0 |
| P4: PRO(X, Z:BAD) | 2 | 0 |
| P5: PRO(Y, Z:GOOD) | 1 | 1 |
| P6: PRO(Y, Z:BAD) | 1 | 2 |
| S1: SUP(Z, X) | 1 | 2 |
| S2: SUP(Z, Y) | 1 | 2 |
| S3: SUP(X, Z:GOOD) | 1 | 1 |
| S4: SUP(X, Z:BAD) | 0 | 0 |
| S5: SUP(Y, Z:GOOD) | 1 | 1 |
| S6: SUP(Y, Z:BAD) | 0 | 0 |
| E1: Subset_of X | 1 | 2 |
| E2: Subset_of Y | 0 | 0 |
| E3: Z PRO (C) | 3 | 1 |
| E4: ANA: (X PRO Y‘) | 0 | 0 |
| E5: ANA: (Z, X or Y) | **4** | **5** |
| OTHER | 2 | 3 |

Table 5: Frequency distribution of question predicates associated with content predicate C2: SUP(X,Y)

| Question | Annotator 1 | Annotator 2 |
|---|---|---|
| P1: PRO(Z, X) | 0 | 1 |
| P2: PRO(Z, Y) | **2** | **2** |
| P3: PRO(X, Z:GOOD) | 0 | 1 |
| P4: PRO(X, Z:BAD) | 1 | 2 |
| P5: PRO(Y, Z:GOOD) | 0 | 0 |
| P6: PRO(Y, Z:BAD) | 1 | 0 |
| S1: SUP(Z, X) | 1 | 2 |
| S2: SUP(Z, Y) | **2** | **3** |
| S3: SUP(X, Z:GOOD) | 2 | 0 |
| S4: SUP(X, Z:BAD) | 1 | 0 |
| S5: SUP(Y, Z:GOOD) | 0 | 1 |
| S6: SUP(Y, Z:BAD) | 0 | 0 |
| E1: Subset_of X | 0 | 0 |
| E2: Subset_of Y | 1 | 1 |
| E3: Z PRO (C) | 0 | 1 |
| E4: ANA: (X PRO Y‘) | 0 | 0 |
| E5: ANA: (Z, X or Y) | 1 | 2 |
| OTHER | 0 | 0 |

Table 6: Frequency distribution of question predicates associated with content predicate C3: ANA(X,Y)

| Question | Annotator 1 | Annotator 2 |
|---|---|---|
| P1: PRO(Z, X) | 0 | 0 |
| P2: PRO(Z, Y) | 0 | 0 |
| P3: PRO(X, Z:GOOD) | 0 | 0 |
| P4: PRO(X, Z:BAD) | 0 | 0 |
| P5: PRO(Y, Z:GOOD) | 0 | 0 |
| P6: PRO(Y, Z:BAD) | 0 | 0 |
| S1: SUP(Z, X) | 0 | 0 |
| S2: SUP(Z, Y) | 0 | 0 |
| S3: SUP(X, Z:GOOD) | 0 | 0 |
| S4: SUP(X, Z:BAD) | 0 | 0 |
| S5: SUP(Y, Z:GOOD) | 0 | 0 |
| S6: SUP(Y, Z:BAD) | 0 | 0 |
| E1: Subset_of X | 0 | 0 |
| E2: Subset_of Y | 0 | 0 |
| E3: Z PRO (C) | 0 | 0 |
| E4: ANA: (X PRO Y‘) | 0 | 0 |
| E5: ANA: (Z, X or Y) | **3** | **2** |
| OTHER | 1 | 1 |