A Language for Graphs of Interlinked Arguments

by Dimitra Zografistou, Giorgos Flouris, Theodore Patkos, and Dimitris Plexousakis (ICS-FORTH)

ArgQL is a high-level declarative language, aimed to query data which are structured as a graph of interconnected arguments. It provides specially designed constructs and terminology that generate queries relevant to the domain of interest that are both easy to express and understand.

The recent advances in the technologies of Web 2.0 changed the role of its users from passive information consumers to active creators of digital content. Web became a universal terrain, wherein humans accommodate their inherent need for communication and self-expression. From a scientific point of view, this new era was accompanied by numerous new challenges. Navigation in dialogues and investigation of the informational requirements is one such challenge, which constitutes a pristine and until recently, almost untouched area. The process of human argumentation, in contrast, has been a longstanding subject of theoretical studies.

Computational argumentation is a branch of AI and it offers more accurate and realistic reasoning methods by transferring the cognitive behaviour of people when arguing into its computational models. An extensive overview in the area led us to the observation that it also defines solid and discrete constructs that structure a dialogue. This observation motivated us to develop ArgQL (Argumentation Query Language) [1, 2], a novel, high-level declarative query language that will allow for the navigation and information identification in a graph of interconnected arguments, structured in the principles of argumentation. ArgQL constitutes an initial effort to understand the informational and theoretical requirements during this process. The most significant contribution lies in its potential to provide a querying mechanism, focused on the internal structures of arguments and their interactions, isolating the process from technical details related to the traditional languages. Its need is highlighted by the complexity of constructing SPARQL queries, even for simple statements, like "How an argument with a given conclusion is attacked?" in the argumentation domain. Instead, ArgQL generates quite elegant and representative queries, easy to both express and understand.

ArgQL was designed to cover several predefined informational requirements, which can be categorised as follows:

- *Individual arguments identification:* We provide features that allow to add constraints to the argument's internal structure, based on particular values.
- *Correlated arguments identification:* ArgQL also allows constraints to be expressed on an argument's content with regard to other arguments, such as: search for pairs of arguments with commonalities in their content.
- *Argument relations extraction*: ArgQL offers built-in keywords that allow express restrictions to be expressed about the relations between arguments.
- *Dialogue traversing and sub-dialogues identification:* Expressions used for navigating across the relation between arguments are also provided.

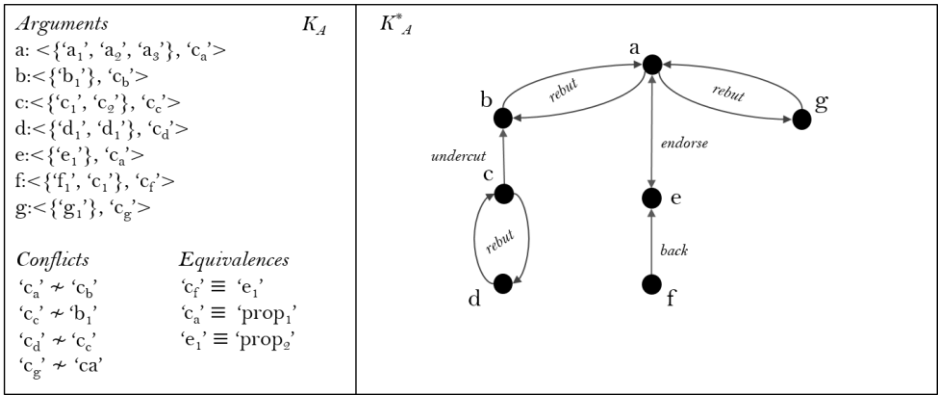

Figure 1: Example of target data.

In figure 1, we show an example of the target data. In the left part, we show the data structures in the lowest level, consisting of arguments and two types of relations between propositions, conflicts and equivalences, while the right depicts their abstract view, in which data form a graph of interlinked arguments.

Below we show two examples of ArgQL, along with an intuitive description of the requirements captured by each.

"Find pairs of arguments with intersected premises, that is, arguments with at least one common premise."

```
Q:  match ?arg1:<?pr1, ?c1> , ?arg2: <?pr2[.?pr1], ?c2>
    return ?arg1, ?arg2
```

"Find all those arguments that attack arguments with conclusion 'c1'".

```
Q:  match ?arg1 attack ?arg2: <?pr, 'c1'>
    return ?arg1
```

As a first step, we formally define a data model based on the prevailing concepts in the area of computational argumentation. Afterwards, we define the language specifications in terms of its syntax, as well as its formal semantics that show how the different keywords and expressions are evaluated against the proposed data model. For query execution, we propose a methodology to translate ArgQL into already well-known storage schemes and in particular the RDF/SPARQL language. The methodology includes the mapping between the data models and the translation between the query languages. The correctness of the translation has been formally proven. We have implemented ArgQL and have also developed an endpoint, wherein queries can be executed against real datasets. The performance of the translation is experimentally evaluated on these datasets. Despite its theoretical correctness, the proposed translation revealed some issues at implementation time, which concerned particular query cases. To address those issues, we suggest a set of optimizations, that result to shorter and, therefore, more effective queries.

References:
[1] D. Zografistou, G. Flouris, D. Plexousakis: "Argql: A declarative language for querying argumentative dialogues", in International Joint Conference on Rules and Reasoning (pp. 230-237), Springer, 2017.
[2] D. Zografistou, et al.: "Implementing the ArgQL Query", Computational Models of Argument, Proc. of COMMA 2018, 305, p.241, 2018.

Please contact:
Dimitra Zografistou, ICS-FORTH, Greece,
+30 2810 391683, dzograf@ics.forth.gr