

NEXAS: A Visual Tool for Navigating and Exploring Argumentation Solution Spaces

Raimund DACHSELT ^{a,d,e}, Sarah Alice GAGGL ^b, Markus KRÖTZSCH ^c,
Julián MÉNDEZ ^a, Dominik RUSOVAC ^b and Mei YANG ^a

^a *Interactive Media Lab Dresden, Technische Universität Dresden*

^b *Logic Programming and Argumentation Group, Technische Universität Dresden*

^c *Knowledge-Based Systems Group, Technische Universität Dresden*

^d *Centre for Tactile Internet (CeTI), Technische Universität Dresden*

^e *Cluster of Excellence Physics of Life (PoL), Technische Universität Dresden*

ORCID ID: Raimund Dachsel <https://orcid.org/0000-0002-2176-876X>, Sarah Alice

Gaggl <https://orcid.org/0000-0003-2425-6089>, Markus Krötzsch

<https://orcid.org/0000-0002-9172-2601>, Julián Méndez

<https://orcid.org/0000-0003-1029-7656>, Dominik Rusovac

<https://orcid.org/0000-0002-3172-5827>

Abstract. Recent developments on solvers for abstract argumentation frameworks (AFs) made them capable to compute extensions for many semantics efficiently. However, for many input instances these solution spaces can become very large and incomprehensible. So far, for the further exploration and investigation of the AF solution space the user needs to use post-processing methods or handcrafted tools. To compare and explore the solution spaces of two selected semantics, we propose an approach that visually supports the user, via a combination of dimensionality reduction of argumentation extensions and a projection of extensions to sets of accepted or rejected arguments. We introduce the novel web-based visualization tool NEXAS that allows for an interactive exploration of the solution space together with a statistical analysis of the acceptance of individual arguments for the selected semantics, as well as provides an interactive correlation matrix for the acceptance of arguments. We validate the tool with a walk-through along three use cases.

Keywords. abstract argumentation, visualization, solution space exploration

1. Introduction

Abstract argumentation is a very active research field within argumentation theory. Besides the theoretical analysis and formal approaches [1] the development of efficient solvers has gained much attention [2], as also witnessed by the International Competition on Computational Models of Argumentation (ICCMA) [3]. This led to a number of solvers that can enumerate all extensions of most of the prominent argumentation semantics. However, the solution space for many semantics can become very large, in particular if many cycles of even length are contained in the given argumentation framework (AF).



Figure 1. NEXAS interface showing a) the *Argument View*: bar chart with the relative frequencies of the arguments by semantics, b) the *Extension View*: scatterplot with the dimensionality reduced solution space, c) the *Correlation View*: correlation matrix of accepted arguments wrt. two semantics separated by the diagonal, and d) a sidebar with the color legend, settings and inspection options.

While theoretical results on AFs already provide a rather clear picture of the relations between the argumentation semantics, for example in terms of subset-relations or the existence of solutions for specific classes of AFs, there is no practical support to explore the solution space of AFs. Standard AF solvers are mainly targeted to i) compute extensions or decide for credulous and skeptical acceptance of arguments efficiently, or ii) to support the construction of arguments and AFs visually [4]. Up to now, the only combination of computational argumentation and visualization is by tools that use a solver in the backend for the computation of solutions or acceptance of arguments, and the visualization of the AF as a directed graph. The outcome of the computation is then visualized by highlighting arguments that are credulously or skeptically accepted, or one extension after another is highlighted in the graph for a selected semantics¹ [5]. With the current tool support the user does not get the full picture on the acceptance of individual arguments, or sets of arguments. When comparing two semantics, handcrafted tools are still needed to post-process all extensions and find out where they coincide or differ. Getting an overview on the whole solution space for specific semantics, to be able to zoom-in or -out into particular sub-spaces that contain similar extensions, is not at all supported by current tools.

To overcome these limitations, we embarked on an interdisciplinary endeavour proposing to utilize advanced visualization to interactively explore the solution space of AFs. As a result, we introduce the NEXAS tool for Navigating and Exploring Argumentation Solutions. In this work we combine the well known Answer Set Programming (ASP) based AF solver `aspartix` [6], and the recently developed approach for answer set navigation [7] to, on one side enumerate extensions, and on the other side navigate towards “interesting” sub-spaces of arguments.

Our main contributions are: (1) An approach that seamlessly integrates visualization techniques and systems for enumerating extensions for abstract argumentation frameworks. (2) Our novel web-based tool NEXAS, an interactive solution that visually realizes the aforementioned approach [7] through three major coordinated views fostering navigation, exploration and understanding of the AF (See Figure 1). (3) Walk-through

¹ Web interface of ConArg (https://conarg.dmi.unipg.it/web_interface.php)

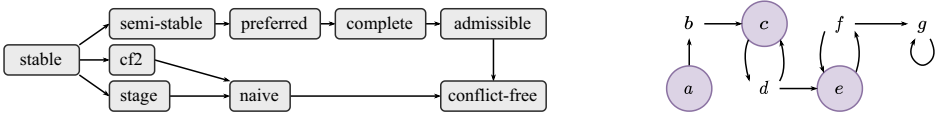


Figure 2. a) (left) An arrow from semantics σ to semantics τ encodes that each σ -extension is also a τ -extension. b) (right) Example AF with highlighted preferred extension $\{a, c, e\}$.

validation of the tool NEXAS on the basis of three use cases. A demo of the tool, video walk-through and test data can be found at <https://imld.de/nexas>.

2. Background and Related Work

2.1. Abstract Argumentation Frameworks

We recall the basic definitions for abstract argumentation frameworks. For more details we refer to the standard literature [8,9].

Definition 1 An abstract argumentation framework (AF) is a pair denoted as $F = (A, R)$ where A is a finite set of arguments and $R \subseteq A \times A$ is the attack relation.

Given a pair $(a, b) \in R$, we say a attacks b . An argument $a \in A$ is defended by a set $S \subseteq A$ if, for each $b \in A$ such that $(b, a) \in R$, there exists a $c \in S$ such that $(c, b) \in R$. Additionally, the range of S (w.r.t. R) is defined as $S_R^+ = S \cup \{x \mid \exists y \in S \text{ such that } (y, x) \in R\}$.

Definition 2 Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is conflict-free (in F), if there are no $a, b \in S$, such that $(a, b) \in R$. $cf(F)$ denotes the collection of conflict-free sets of F . For a conflict-free set $S \in cf(F)$, it holds that

- S is an admissible set of F , i.e., $S \in adm(F)$, if each $a \in S$ is defended by S ;
- S is a stable extension of F , i.e., $S \in stb(F)$, if $S_R^+ = A$;
- S is a complete extension of F , i.e., $S \in compl(F)$, if $S = \{s \in A \mid S \text{ defends } s\}$;
- S is a preferred extension of F , i.e., $S \in prf(F)$, if $S \in compl(F)$ and there is no $T \in compl(F)$ with $T \supset S$;
- S is a stage extension of F , i.e., $S \in stg(F)$, if there is no $T \in cf(F)$, such that $S_R^+ \subset T_R^+$;
- S is a maximal conflict-free or naive set of F , i.e., $S \in naive(F)$, if $S \in cf(F)$ and for each $T \in cf(F)$, $S \not\subset T$;
- S is a semi-stable extension of F , i.e., $S \in semis(F)$, if and only if $S \in compl(F)$ and there is no $T \in compl(F)$, such that $S_R^+ \subset T_R^+$.

Our tool also supports the SCC-recursive semantics $cf2$. For more details we refer to the following article [9]. Figure 2a) shows the relation between the introduced semantics in terms of subset-relations. Figure 2b) shows an example AF represented as a directed graph where the preferred extension $\{a, c, e\}$ is highlighted.

We denote the set of arguments of an AF F that are credulously accepted under semantics σ by $cred_\sigma(F) := \bigcup \sigma(F)$, and the set of arguments that are skeptically accepted under semantics σ by $skep_\sigma(F) := \bigcap \sigma(F)$.

2.2. Related Work

Visualization in Abstract Argumentation Frameworks. Although there are many argumentation systems [2], approaches that concentrate on the visualization of solution spaces for AFs have not yet been investigated in detail. The only work we are aware of is the *Neva* tool, a preliminary version of the NEXAS tool presented in [10].

The standard way to visualize extensions wrt. a semantics of an AF is by highlighting the accepted arguments in a directed graph, as shown in Figure 2b). In the following we discuss also related work on visualization for related domains, such as ASP. For example, *ARVis* [11] visualizes answer sets and their relations using a directed graph structure. Besides, the toolkit *Possible Worlds Explorer* (PWE) [12] is able to visualize individual answer sets and their structures. Additionally, PWE builds a distance matrix based on the similarity measure and applies hierarchically clustering on it to discover further substructures. On the other side, Betz et al. [13] implemented a tool based on argumentation theory that can structure and visualize multi-dimensional opinions by mapping them into two-dimensional space. Each opinion is a feature of the structured AF. They also cluster the opinion vectors (argumentation extensions) and calculate the coherence of opinion vectors. However, it is only suitable for very small data sets and focuses on the data relations.

Dimensionality Reduction. Dozens of dimensionality reduction techniques (also known as projections) are currently available and most are summarized in the surveys by Esposito et al. [14] and Nonato et al. [15]. They are mainly used for exploration of multidimensional data, notably in AI applications and information visualization (InfoVis). For example, to explore gene expression patterns and correlate the results with classically defined neuroanatomy, Ji [16] projects hybridization gene expression data into a two-dimensional space using t-SNE and PCA algorithms and visualizes it using scatterplots. Among these techniques, the variants of MCA (*Multiple Correspondence Analysis*) [17,18] are among the few techniques capable to encoding categorical data to a visual space such that closeness reflects similarities [15].

Visualization of Highly Dimensional Data. From InfoVis, there are techniques tailored to highly dimensional data [19,20], like Parallel Coordinates, Projection Views, Shape Coding, Recursive Pattern, etc. However, displaying a large amount of data objects with many features remains an open challenge where, usually, no single technique suffices. For this problem, tailoring multiple linked views has been proven to be effective [20], while also enabling the design of interactions and interplay between the views towards specific tasks. These design possibilities have recently been used to deal with specific use cases in various AI and formal methods domains (e.g., [21,22]). Furthermore, the visual complexity of the visualization plays an important role for explainability in AI, as expressed in the user needs of the What-If-Tool [23]. In fact, this tool successfully employs scatterplots, bar charts and line charts to provide insights about the model, data and features involved, when as stated before, one could opt for more intricate visualization techniques to deal with the complexity of the data.

Set Visualization. For the specific case of set-typed data, visualization techniques have also been studied well [24]. For example, based on Euler diagrams, *EULERFORCE* [25] uses a force-directed layout to optimize space usage, and *SPEULER* [26] includes semantic information in a two step layout method for highly overlapping sets. *ONSET* [27] uses

a pixel-oriented technique to show which elements are included in the sets represented with juxtaposed grids, in contrast to the SET´O´GRAM approach [28], which uses superimposed bar charts to obtain insights. Other approaches use multiple coordinated views, like POWERSET [29], which uses bar charts and treemaps to show set intersections. Similar open challenges about the amount of data that can be visualized at once also apply to set visualization techniques, specially when many sets and their contents need to be compared. Thus, we chose familiar and effective representations that ease the cognitive load for users at different levels of expertise, as most of the existing work is best (or only) suited for small datasets.

3. NEXAS: Navigating and Exploring Argumentation Solution Spaces

So far, there is no tool that supports the solution space exploration for AFs. This means that in case there are hundreds or thousands of solutions, there is no way to see where for example two semantics coincide, or which arguments are accepted often or rarely.

Use Cases. In the following we describe typical Use Cases (UC) where the existing tool support is not enough. As a user for such scenarios we consider either i) a researcher that works in the field of abstract argumentation who uses argumentation systems to investigate certain aspects of argumentation semantics, or ii) a student in abstract argumentation who attends a course on argumentation theory and wants to learn more about the behavior of argumentation semantics. Thus, the tool is aimed to support the user visually in the analysis of the outcome on the computation of the argumentation semantics. The following use cases will help us to identify the goals on the visualization tool.

UC-1: The user wants to explore the AF solution space (the set of extensions) for specific argumentation semantics of a given AF. The main goal is to compare the semantics and to find out about similarities and differences. a) The individual acceptance rate of each argument should be given. The user wants to see the relative frequency of acceptance wrt. the semantics for each argument, and for the intersection of the extensions of both semantics. b) She wants to be able to select arguments and see in which extensions they are accepted. It is also desirable to be able to select several arguments and get the union of solutions highlighted where they are accepted. c) Each individual extension should be observable, and a representation that visualizes the complete solution space of two selected semantics is needed, which allows to identify which extensions share many arguments, and which have less in common.

UC-2: The solution space for one or both semantics might just be too big to be computed. On one side due to the time and space needed for the computation and on the other side because the information which needs to be processed is too big to be comprehensible. In most applications the user might just want to compute a particular subspace of the whole solution space, where some arguments are either contained in all or in none of the extensions.

UC-3: Besides highlighting the extensions where particular arguments are accepted (see UC-1), the user wants to “navigate” towards desired solution sub-spaces, by choosing which arguments should be accepted or rejected. To this end, the tool needs to show for which arguments such a selection is even possible. Those are arguments that are cred-

ulously but not skeptically accepted. Note that choosing skeptically accepted arguments would not change the solution space.

Design Goals. Within interdisciplinary brainstorming and feedback sessions involving experts in *Abstract Argumentation* as well as *Interactive Visualization*, we iteratively designed our NEXAS tool. Within these conversations, we considered needs of the argumentation community which resulted in developing a set of design goals (DG) that our tool must fulfill.

DG-1: Intuitive and Familiar Representations. We aim to foster intuitive understanding of the views by using traditional representations of the AF components while also encoding relevant information that the users can obtain insights from.

DG-2: Highlight Component Relations. A major challenge is to understand how components affect others. Thus, we aim to make these relations visible through linked interactions to foster understanding of the underlying framework.

DG-3: Maintainable and Customizable. The system design must be flexible and allow incorporation of further components in future iterations.

DG-4: Support Several Tasks and Workflows. We aim to support tasks with disjoint purposes and thus the available interactions must reflect such purposes.

DG-5: Ready-to-use. We aim to minimize setup complexity of the tool to account for various user environments.

4. Technical Design and Implementation

Figure 3 provides a simplified version of steps the data goes through in our system, also indicating the relationships between views.

Preprocessing Extensions. The backend uses the python library `pandas`² to perform operations on datasets during preprocessing. After computing the semantics using the ASP solver, we use a *one-hot-encoding* in order to construct two binary datasets D and D_s from the obtained extensions (answer sets). D simply encodes extensions as rows where columns correspond to arguments that belong to at least one extension (credulously accepted arguments) whose respective values $v \in \{0, 1\}$ indicate whether they are contained ($v = 1$) in the respective extensions, or not ($v = 0$). The other dataset D_s additionally contains flags for whether an extension is contained in one semantics, the other semantics or in both (the intersection) of them.

Dimensionality Reduction. Out of the presented reduction methods, t-SNE and MCA seem to be best suited for our application. Due to performance reasons we decided to use MCA in our tool. The two-dimensional reduction by MCA (*Multiple Correspondence Analysis*) using the python package `prince`³ is performed on D and D_s , respectively. Ultimately the two-dimensional reduction provides the coordinates for the scatterplot that visualizes extensions. For D the dimensionality reduction is based on which arguments an extension contains, hence, the reduction reflects the similarity of extensions. For D_s the reduction in addition takes into account to which semantics an extension belongs. Performing a reduction on the latter data set may thus cause more distinctive clusters of data points in the scatterplot.

² <https://pandas.pydata.org/> ³ <https://github.com/MaxHalford/prince>

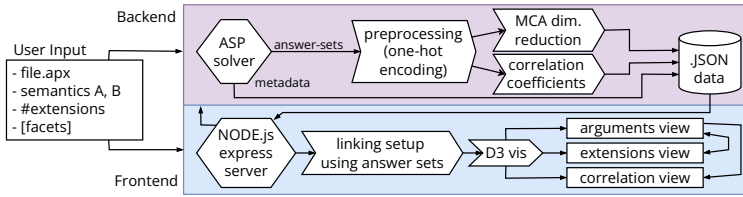


Figure 3. System diagram, describing our data management to build the visualizations.

Correlation. We compute correlation coefficients of arguments using the `pandas` library on dataset D , which simply reflects the containment of arguments among extensions. Datasets containing correlation coefficients are computed for each semantics, respectively, providing the frontend with data that can be used to visualize correlations in each of the semantics separately.

Navigating Extensions. We use the recently proposed ASP navigation framework [7] for navigating extensions of AFs. The main idea is to understand arguments that are credulously, but not skeptically accepted, as so called *facets* of the respective semantics of an AF, which can be included or excluded from the solution space in order to land in a sub-space of extensions. So called navigation steps are performed by inclusion or exclusion of facets. The inclusion of a facet prunes each extension that does not contain the corresponding argument, and accordingly, the exclusion of a facet prunes each extension that contains the corresponding argument. Navigation steps are achieved by adding corresponding integrity constraints to the respective answer set program that encodes the AF. By recursively enforcing that arguments are included in or excluded from extensions (answer sets), we can zoom into or out of the solution space, which ultimately provides us with functionality to navigate extensions. For more details on weighted faceted answer set navigation, interested readers are referred to [7].

Faceted navigation is realized visually in the frontend, using the one-hot-encoded data from the backend to perform navigation steps and display the resulting sub-spaces (addressing design goal DG-2). Furthermore, users can provide a list of facets to activate at startup (UC-2), to directly land in the resulting sub-space, which may be useful for very large solution space, as users can prune certain extensions right in the beginning.

Implementation Challenges. Several steps within the tool are computationally very expensive. This starts with the enumeration of extensions for the argumentation semantics, and continues with the processing of the data with dimensionality reduction and the visualization. Therefore, we preferred methods that are 1) efficient, 2) robust, 3) easy to adapt and 4) to combine (DG-3). To this end, we decided to use the `aspartix` [6] approach for the enumeration of extensions. Within the `aspartix` system suite, there are ASP encodings for most of the prominent semantics, and these encodings can be easily extended or modified independently from our tool NEXAS. As the `aspartix` approach relies on ASP technology, we can adapt the answer set navigation approach from [7] to allow a more interactive and informative exploration of the AF solution space. On the visualization front, to simplify access to the tool considering the work environments of our target users (DG-5), we provide NEXAS as a web application which—since it is limited by the resources of the browser tab—needs to avoid expensive operations and update the website components efficiently.

Data Generation. To generate the answer sets and miscellaneous data, we use a python script that calls the ASP solver `clingo` [30] together with the ASP encodings of `aspartix` [6] and the ASP solution space navigation approach [7]. For the computation of the semantics pairs we can often make use of the theoretical knowledge about the subset-relation between the semantics (see Figure 2). This means that for the pairs of semantics $\text{stable} \subseteq \sigma$ for $\sigma \in \{\text{semi-stable, stage, preferred, cf2}\}$, we only compute the extensions of the super-set, and identify within the answer sets obtained, which of the solutions are also an extension of the other semantics. The answer sets are finally stored in `.json` files for the frontend to use.

Web Application. NEXAS has a server implemented with `express.js`⁴. The HTML content is built using the templating engine `sprightly`⁵ to facilitate component-based UI that can be easily maintained (DG-3). On the client side, the look and feel is achieved by adapting `Materialize`⁶ components. The visualizations are built using `D3`⁷, since it is flexible enough to support non-standard visual encodings.

5. Visualization Design

The NEXAS tool consists of 3 major views to explore and navigate the solution space of an abstract argumentation framework. All views can be seen on Figure 1 and their influence on each other is encoded with arrows in Figure 3. The semantics are color encoded in all views (DG-2) using contrast to distinguish between them easily.

Argument View. This view encodes the set of arguments (Figure 1 a)). A double vertical bar chart is used to compare, for each argument, the relative frequency with which it is accepted in each of the two semantics (A, B). With a central axis used to distinguish between the semantics, we show A in the bars to the left and B in the bars to the right. The intersection of the semantics (i.e., the frequency with which an argument is accepted in both A and B) is shown as a hatching pattern on both sides of the bar chart. The width of the intersection hatching can cover 100% of both sides of the chart, to indicate whether the intersection subsumes the solutions for either semantics. With this, we show at a glance how the semantics compare (DG-1). This bar chart can be used to filter and highlight content in the other views.

Extensions View. The solution space (i.e., set of extensions) is displayed in a scatterplot (Figure 1 b)). The extensions that belong either to A or B are encoded with the same colors as in the bar chart, and those that belong to both A and B are also encoded with a third color that visually blends the colors used for the two semantics. The MCA dimensionality reduction provides a spatial clustering based on the similarity of the contents of each solution, which also enables users to immediately get a feeling of how the extensions are distributed by semantics (DG-1). To show when several solutions are stacked on top of each other, we lowered the opacity of each point in the scatterplot. Furthermore, it is possible to include the semantics in the dimensionality reduction from the settings sidebar, emphasizing the grouping by semantics from the MCA. Users can also inspect the arguments contained in each solution by selecting them, which triggers highlights in the Argument View. This effect can be seen in Figure 4 b).

⁴ <https://expressjs.com/>

⁵ <https://www.npmjs.com/package/sprightly>

⁶ <https://materializecss.com/>

⁷ <https://d3js.org/>

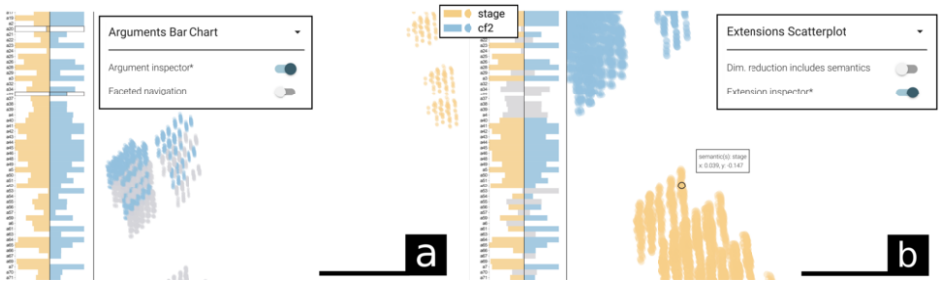


Figure 4. NEXAS interface showing a) the selection of a20 and 36 together with the extensions highlighted in the scatterplot in which they are accepted, b) the selection of one extension in the scatterplot together with the arguments highlighted in the argument bar char that are contained in it.

Correlation View. Figure 1 c) displays the correlation matrix split across the diagonal to compare the values in both semantics: A on the upper triangular matrix and B on the lower one. The semantics are shown in the color coding of the perimeter of both triangles (DG-2). Since the correlation matrix of a single semantics is symmetric, no information is lost. Additionally, hovering over any cell on the matrix highlights the same argument pair on the other side of diagonal, so that users can compare the correlations values (from -1 to 1) on demand. The vertical axis of the matrix of this view and bar chart of the Argument View correspond, such that hovering over an argument in the bar chart adds a visual guide over the row and column of the highlighted argument in the matrix.

Cross-view Interactions. We provide multiple ways of creating a selection of either arguments or extensions. We distinguish these as *Argument Inspector* and *Extension Inspector*, of which only one may be active at a time (DG-4). The user selection can be exported at any time depending on the active inspector. While the Argument Inspector is on, users may select arguments on the bar chart to highlight the extensions on the scatterplot that contain the selection (Figure 4 a)). For the Extension Inspector, the selection is done on the scatterplot and the result is reflected on the bar chart (Figure 4 b)).

On the other hand, the *Faceted Navigation* described in Section 3 can be enabled from the settings sidebar and alters the behavior of both the Argument and Extension Inspectors (to support UC-3). When active, the arguments which are not facets of the remaining extension set (i.e., the arguments are either contained in all solutions or none) are disabled. The Argument Inspector then allows users to include facets (click) or exclude them (ctrl + click), which updates the solution space by filtering the extensions accordingly. Multiple *simultaneous* selections are disallowed in Faceted Navigation, as after including or excluding a facet, the remaining arguments may not be facets of the resulting solution space. However, users can choose multiple facets *sequentially*, as the available facets update after each selection. Likewise, the Extension Inspector only allows one selected solution at a time while Faceted Navigation is active. For visual reference, see Figure 5.

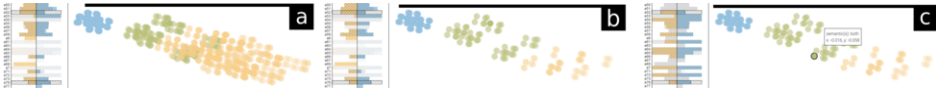


Figure 5. Faceted Navigation with a) the pre-selection of a54 as Preset Facet, and included facets a52, a21 and a76, b) after also including facet a94, and c) inspecting one extension using the *Extension Inspector*.

6. Validating NEXAS

Here we demonstrate the feasibility of our approach by means of a walk-through inspired by the use cases from Section 3. Initially the user provides the input AF⁸. All figures of the NEXAS interface use this input file. The upper bound⁹ for the number of extensions to be computed for each of the semantics, respectively, is set to 10,000. Further, semantics stage and cf2 are selected for the comparison.

After the data generation, the user sees the views as shown in Figure 1. As requested in UC-1 a) the argument view reveals that the semantics have no extension in common, as for none of the arguments there appears an overlay of stripes. When hovering over argument a54, a box appears and provides the information that it is accepted in 33.2% of the stage extensions and in 25.92% of the cf2 extensions. Further, one can observe that for instance argument a32 is solely accepted in 50% of the stage extensions, and not for cf2 semantics. Following UC-1 b) the user clicks on a20 and observes in the scatterplot that only the stage extensions, that contain a20, are displayed in color, all other points are grayed out. By further clicking on a36, additionally several cf2 extensions are again displayed in color, thus showing all extensions where a20 or a36 is accepted. After clearing the selection, the user switches to the Extension Inspector and clicks on one extension in the scatterplot. This made all arguments, that are not contained in that particular extension, grayed out in the argument bar chart, and thus all arguments that are still colored are accepted in that extension (UC-1 c)). Selecting another extension nearby the previous one reveals, as mentioned in UC-1 c), that extensions that are placed near to each other in the scatterplot share many arguments.

Considering UC-2, the user is interested in exploring the sub-space of solutions where argument a54 is always accepted. She has two options, either to use faceted inspection, which then only considers the sub-space restricted to a maximum of 10,000 for each of the semantics, or to restart the tool with a modified configuration where a54 is activated as an inclusive facet so that all computed extensions contain a54. Using the second option, leads to sub-spaces that have 432 extensions in common. The argument a32, which previously was accepted in 50% of the stage extensions, is now accepted in 45.68%. The user selects a52, a21 and a76 in Faceted Navigation (UC-3), leading to the state visible in Figure 5 a). By further including facet a94, the number of extensions is reduced again as shown in Figure 5 b). Note that Faceted Navigation zooms in the solution space, thus all extensions that don't contain a52, a21, a76 and a94 are not displayed any longer, in contrast to the default behavior of the Argument Inspector that would gray out the other extensions. Next, the user observes one particular extension by switching to the Extension Inspector (while still having Faceted Navigation on), and clicking on an extension in the scatterplot. Then, in the argument bar chart on the left, all arguments are highlighted in color which are accepted in that extension (Figure 5 c)).

⁸ massachusetts_sрта_2014-11-13.gml.50.apx from the instances of ICCMA 2017 [31].

⁹ Setting an upper bound for extensions could result in only a part of the solution space being considered.

7. Conclusion and Future Work

With our visual analysis tool NEXAS we propose a novel way of exploring the argumentation solution space, that offers a tight integration of state of the art solvers for AFs and interactive visualization design. Our validation shows that a user can easily observe information that, without the tool, would be very hard to comprehend, such as the relative frequency of acceptance of arguments, or which extensions belong to both semantics under consideration. The linked interactions within the three seamlessly integrated visualization views allow for multiple and advanced ways of solution space exploration.

For future work we will further refine and expand NEXAS, by integrating weights for facets, as described in [7]. Moreover, we will investigate how to represent the input AF in NEXAS, and how to integrate it in the views of the tool. User studies shall help us to integrate feedback from experts in the field of abstract argumentation. We believe that NEXAS opens up new ways to easily explore solution spaces, which will pave the way to also lift concepts from NEXAS to more general settings like ASP solution spaces.

8. Acknowledgements

This work was funded by the BMBF in projects 01IS20056_NAVAS and ScaDS.AI (Center for Scalable Data Analytics and Artificial Intelligence), by the Center for Advancing Electronics Dresden (cfaed), and by the DFG under Germany's Excellence Strategy: EXC-2068, 390729961 — Cluster of Excellence "Physics of Life" and EXC 2050/1, 390696704 — Cluster of Excellence "Centre for Tactile Internet" (CeTI) of TU Dresden, by DFG grant 389792660 as part of TRR 248 – CPEC.

References

- [1] Baroni P, Gabbay DM, Giacomin M, van der Torre L. Handbook of Formal Argumentation. College Publications; 2018. Available from: https://books.google.de/books?id=_OnTswEACAAJ.
- [2] Cerutti F, Gaggl SA, Thimm M, Wallner JP. Foundations of Implementations for Formal Argumentation. FLAP. 2017;4(8). Available from: <http://www.collegepublications.co.uk/downloads/ifcolog00017.pdf>.
- [3] Lagniez J, Lonca E, Mailly J, Rossit J. Design and Results of ICCMA 2021. CoRR. 2021;abs/2109.08884., doi: 10.48550/arXiv.2109.08884.
- [4] Janier M, Lawrence J, Reed C. OVA+: an Argument Analysis Interface. In: Parsons S, Oren N, Reed C, Cerutti F, editors. Proc. of COMMA 2014. vol. 266 of FAIA. IOS Press; 2014. p. 463-4., doi: 10.3233/978-1-61499-436-7-463.
- [5] Bistarelli S, Rossi F, Santini F. ConArgLib: an argumentation library with support to search strategies and parallel search. J Exp Theor Artif Intell. 2021;33(6):891-918., doi: 10.1080/0952813X.2020.1789756.
- [6] Dvorák W, Gaggl SA, Rapberger A, Wallner JP, Woltran S. The ASPARTIX System Suite. In: Prakken H, Bistarelli S, Santini F, Taticchi C, editors. Proc. of COMMA 2020. vol. 326 of FAIA. IOS Press; 2020. p. 461-2., doi: 10.3233/FAIA200534.
- [7] Fichte JK, Gaggl SA, Rusovac D. Rushing and strolling among answer sets—navigation made easy. In: Proc. of the AAAI 2022. vol. 36; 2022. p. 5651-9., doi: 10.1609/aaai.v36i5.20506.
- [8] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence. 1995;77(2):321-57., doi: 10.1016/0004-3702(94)00041-X.
- [9] Baroni P, Caminada M, Giacomin M. An introduction to argumentation semantics. The knowledge engineering review. 2011;26(4):365-410., doi: 10.1017/S0269888911000166.

- [10] Yang M, Gaggl SA, Rudolph S. Neva - Extension Visualization for Argumentation Frameworks. In: Prakken H, Bistarelli S, Santini F, Taticchi C, editors. Proc. of. COMMA 2020. vol. 326 of FAIA. IOS Press; 2020. p. 477-8., doi: 10.3233/FAIA200542.
- [11] Ambroz T, Charwat G, Jusits A, Wallner JP, Woltran S. ARVis: Visualizing relations between answer sets. In: Cabalar P, Son TC, editors. Proc. of LPNMR 2013. Springer. Springer; 2013. p. 73-8., doi: 10.1007/978-3-642-40564-8_8.
- [12] Gupta S, Cheng YY, Ludäscher B. Possible Worlds Explorer: Datalog & Answer Set Programming for the Rest of Us. In: Datalog; 2019. p. 44-55.
- [13] Betz G, Hamann M, Mchedlidze T, von Schmettow S. Applying argumentation to structure and visualize multi-dimensional opinion spaces. *Argument & Computation*. 2019;10(1):23-40., doi: 10.3233/AAC-181004.
- [14] Espadoto M, Martins RM, Kerren A, Hirata NST, Telea AC. Toward a Quantitative Survey of Dimension Reduction Techniques. *IEEE Trans Vis Comput Graph*. 2021;27(3):2153-73., doi: 10.1109/TVCG.2019.2944182.
- [15] Nonato LG, Aupetit M. Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment. *IEEE Trans Vis Comput Graph*. 2019;25(8):2650-73., doi: 10.1109/TVCG.2018.2846735.
- [16] Ji S. Computational genetic neuroanatomy of the developing mouse brain: dimensionality reduction, visualization, and clustering. *BMC bioinformatics*. 2013;14(1):1-14., doi: 10.1186/1471-2105-14-222.
- [17] Greenacre M, Blasius J. Multiple correspondence analysis and related methods. Chapman and Hall/CRC; 2006., doi: 10.1201/9781420011319.
- [18] Tenenhaus M, Young FW. An analysis and synthesis of multiple correspondence analysis, optimal scaling, dual scaling, homogeneity analysis and other methods for quantifying categorical multivariate data. *Psychometrika*. 1985 Mar;50(1):91-119., doi: 10.1007/BF02294151.
- [19] Liu S, Maljovec D, Wang B, Bremer PT, Pascucci V. Visualizing High-Dimensional Data: Advances in the Past Decade. *IEEE Trans Vis Comput Graph*. 2017;23(3):1249-68., doi: 10.1109/TVCG.2016.2640960.
- [20] Keim DA. Information visualization and visual data mining. *IEEE Trans Vis Comput Graph*. 2002;8(1):1-8., doi: 10.1109/2945.981847.
- [21] Horak T, Coenen N, Metzger N, Hahn C, Flemisch T, Méndez J, et al. Visual Analysis of Hyperproperties for Understanding Model Checking Results. *IEEE Trans Vis Comput Graph*. 2022 10., doi: 10.1109/TVCG.2021.3114866.
- [22] Chegini M, Bernard J, Berger P, Sourin A, Andrews K, Schreck T. Interactive labelling of a multivariate dataset for supervised machine learning using linked visualisations, clustering, and active learning. *Visual Informatics*. 2019;3(1):9-17., doi: 10.1016/j.visinf.2019.03.002.
- [23] Wexler J, Pushkarna M, Bolukbasi T, Wattenberg M, Viégas F, Wilson J. The What-If Tool: Interactive Probing of Machine Learning Models. *IEEE Trans Vis Comput Graph*. 2020;26(1):56-65., doi: 10.1109/TVCG.2019.2934619.
- [24] Alsallakh B, Micallef L, Aigner W, Hauser H, Miksch S, Rodgers P. The State-of-the-Art of Set Visualization. *Computer Graphics Forum*. 2016;35(1):234-60., doi: 10.1111/cgf.12722.
- [25] Micallef L, Rodgers P. eulerForce: Force-directed layout for Euler diagrams. *Journal of Visual Languages & Computing Distributed Multimedia Systems DMS2014 Part I*. 2014;25(6):924-34., doi: 10.1016/j.jvlc.2014.09.002.
- [26] Kehlbeck R, Görtler J, Wang Y, Deussen O. SPEULER: Semantics-preserving Euler Diagrams. *IEEE Trans Vis Comput Graph*. 2022;28(1):433-42., doi: 10.1109/TVCG.2021.3114834.
- [27] Sadana R, Major T, Dove A, Stasko J. OnSet: A Visualization Technique for Large-scale Binary Set Data. *IEEE Trans Vis Comput Graph*. 2014;20(12):1993-2002., doi: 10.1109/TVCG.2014.2346249.
- [28] Freiler W, Matkovic K, Hauser H. Interactive Visual Analysis of Set-Typed Data. *IEEE Trans Vis Comput Graph*. 2008;14(6):1340-7., doi: 10.1109/TVCG.2008.144.
- [29] Alsallakh B, Ren L. PowerSet: A Comprehensive Visualization of Set Intersections. *IEEE Trans Vis Comput Graph*. 2017;23(1):361-70., doi: 10.1109/TVCG.2016.2598496.
- [30] Gebser M, Kaminski R, Kaufmann B, Lühne P, Obermeier P, Ostrowski M, et al. The Potsdam Answer Set Solving Collection 5.0. *Künstliche Intell*. 2018;32(2-3):181-2., doi: 10.1007/s13218-018-0528-x.
- [31] Gaggl SA, Linsbichler T, Maratea M, Woltran S. Design and results of the Second International Competition on Computational Models of Argumentation. *Artif Intell*. 2020;279., doi: 10.1016/j.artint.2019.103193.