The `argumentation` Package

Lars Bengel
lars.bengel@fernuni-hagen.de

November 5, 2023

Contents

1 Example

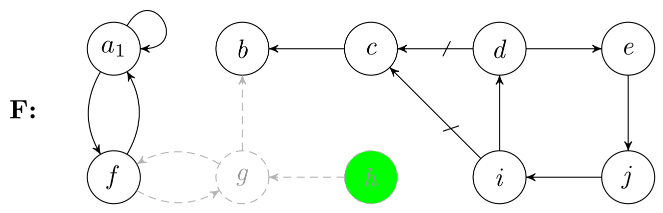


Figure 1: An exemplary AF created with the argumentation package.

```
\usepackage{argumentation}
\begin{figure}[ht]
    \centering
    \begin{af}
        \argument{args1}{a}
        \argument[right=of args1]{args2}{b}
        \argument[right=of args2]{args3}{c}
        \argument[right=of args3]{args4}{d}
        \argument[right=of args4]{args5}{e}
        \argument[below=of args1]{args6}{f}
        \argument[inactive,right=of args6]{args7}{g}
        \argument[inactive,argin,right=of args7]{args8}{h}
        \argument[right=of args8]{args9}{i}
        \argument[right=of args9]{args10}{j}

        \afname[left of=args1,yshift=-0.8cm,xshift=-0.2cm]{cap}{\textbf{F:}}

        \selfattack{args1}
        \dualattack[]{args1}{args6}
        \dualattack[inactive]{args6}{args7}

        \attack[inactive]{args8}{args7}
        \attack[inactive]{args7}{args2}
        \attack[]{args3}{args2}
        \attack[]{args4}{args5}
        \attack[]{args5}{args10}
        \attack[]{args10}{args9}
        \attack[]{args9}{args4}

        \support[]{args4}{args3}
        \support[]{args9}{args3}
    \end{af}
    \caption{An exemplary AF created with the \textsf{argumentation} package.}
    \label{fig:example}
\end{figure}
```

2 Documentation for Version 1.0 [2023/11/05]

In the following, we provide an overview over the functionality of the argumentation package.

2.1 Package Options

Multiple options are provided to customize the look of the argumentation framework. The namestyle option accepts three different values

italics	(default) The argument name is rendered in *italics*.
bold	The argument name is rendered in **bold**.
bolditalics	The argument name is rendered with ***both***.
normal	The argument name is rendered normally.

The argumentstyle option controls the style of the argument nodes and accepts two values

standard	(default) Standard style for the argument nodes.
retro	Alternative style, thicker outline and slightly larger nodes.

The attackstyle option controls the style of the attack arrows and accepts two values

standard	(default) Standard style for the attack arrow tips.
retro	Alternative style, arrow tip is larger and sharper.

The supportstyle option controls the style of the support arrows and accepts three values

standard	(default) Standard style for the attack arrow tips.
dashed	Dashed arrow line, same tip.
double	Double arrow line and large flat tip.

2.2 Environments

The package provides two environments for creating abstract argumentation frameworks and bipolar argumentation frameworks in LaTeX-documents.

2.2.1 af-Environment

The argumentation package provides the af environment for creating abstract argumentation framework. The af environment extends the tikzpicture environment, meaning all tikzpicture-parameters can be used inside the af environment as well. The most relevant parameter is `node distance`, which is set to `1cm` per default.

2.2.2 miniaf-Environment

The miniaf environment can be used to create argumentation frameworks using less space. Especially useful for two-column layout documents. It provides essentially the same as the af environment, but with `node distance=0.5cm` and for each node `minimum size=0.5cm`, `font=\small`.

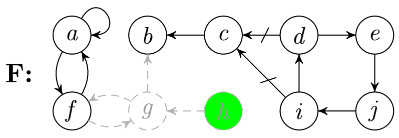


Figure 2: An exemplary Mini-AF created with the miniaf environment.

2.3 Arguments

Arguments can be created with the following command

```
\argument{id}{name}
```

To create an argument, you must provide a unique identifier id and the name to be displayed in the picture. The id of an argument is then referred to when creating attacks as well as for the relative positioning of the other arguments.

The standard style of an argument is defined with the following parameters, all of which can be overridden if desired.

circle	the shape of the argument.
minimum size=0.75cm	the minimum size of the circle, to ensure consistent argument size.
draw=black	outline and text color of the argument.
thick	the outline of the circle is rendered in thick mode.
fill=white	the background color of the argument.
font=large	the font size of the argument name.
text centered	positioning of the argument name inside the circle.
inner sep=0	inner margins of the circle, set to 0 to optimize space.

2.3.1 Relative Positioning

This package supports relative placement of the arguments via the tikz-library positioning. The relative positioning information is provided as an optional parameter via

```
\argument[dir=of arg_id]{id}{name}
```

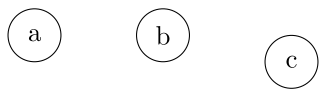
with dir being one of: *right*, *left*, *below* and *above* and arg_id being the id of another argument.

Additionally, you can adjust the horizontal/vertical position of an argument via the options xshift and yshift. You must also specify the distance in one of the following ways

```
\argument[xshift=5mm]{id}{name}
\argument[xshift=5pt]{id}{name}
\argument[xshift=5ex]{id}{name}
```

Example 1

```
\begin{af}
    \argument{arg1}{a}
    \argument[right=of arg1]{arg2}{b}
    \argument[right=of arg2, yshift=-10pt]{arg3}{c}
\end{af}
```



2.3.2 Argument Styles

Furthermore, you can provide optional parameters to adjust the style of the argument node. For that you can use all tikz-style options and additionally the following pre-defined style parameters:

inactive	The argument is displayed in grey and with a dotted outline.
argin	The argument is displayed with green background color.
argout	The argument is displayed with red background color.
argundec	The argument is displayed with cyan background color.

Example 2

2.4 Attacks

Attacks between two arguments can be created with the command

```
\attack{arg1}{arg2}
```

where arg1 and arg2 are the identifiers of two previously defined arguments. The standard style for attacks is defined with the arrows.meta library as follows

```
-Stealth[scale=1.25]
```

2.4.1 Attack Styles

To customize an attack you can provide additional optional parameters:

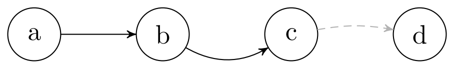
inactive	The attack is displayed in grey and with a dotted line.
bend right	The attack arrow is bent to the right.
	Can additionally provide the angle, e. g., bend right=40.
bend left	The attack arrow is bent to the left. Can also provide an angle.

Furthermore, all tikz style parameters can be used here as well.

Example 3

```
\begin{af}
    \argument{arg1}{a}
    \argument[right=of arg1]{arg2}{b}
    \argument[right=of arg2]{arg3}{c}
    \argument[right=of arg3]{arg4}{d}

    \attack{arg1}{arg2}
    \attack[bend right]{arg2}{arg3}
    \attack[bend left=10,inactive]{arg3}{arg4}
\end{af}
```

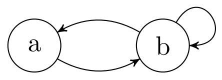


Additionally, you can create a symmetric attack between two arguments with
`\dualattack{arg1}{arg2}`
and a self-attack for an argument with
`\selfattack{arg1}`
For both commands, you can use the same optional parameters as for the `\attack` command.

Example 4

```
\begin{af}
    \argument{arg1}{a}
    \argument[right=of arg1]{arg2}{b}

    \selfattack{arg1}
    \dualattack{arg1}{arg2}
\end{af}
```



2.5 Supports

You can create a support relation between two arguments with the command

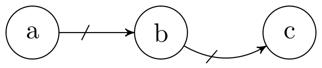
```
\support{arg1}{arg2}
```

where `arg1` and `arg2` are the identifiers of two previously defined arguments. The support arrow use the same tip as the attack arrows, but have a perpendicular mark to distinguish them from attacks. Supports can be customized in the same way as attacks.

Example 5

```
\begin{af}
    \argument{arg1}{a}
    \argument[right=of arg1]{arg2}{b}
    \argument[right=of arg2]{arg3}{c}

    \support{arg1}{arg2}
    \support[bend right]{arg2}{arg3}
\end{af}
```



2.6 Further Commands

If you want to display an identifier for your argumentation framework in the picture, you can use the command

```
\afname{id}{name}
```

where `id` is an identifier for the created node and `name` is the text displayed in the picture. Additionally, positioning information can be provided via the optional parameters.

Example 6

```
\begin{af}
    \argument{arg1}{a}
    \argument[right=of arg1]{arg2}{b}
    \afname[left=of arg1]{caption}{$F:$}

    \attack{arg1}{arg2}
\end{af}
```

$$\begin{tikzcd} F: & a \arrow[r, "/" marking] & b & c \end{tikzcd}$$