

# Knowledge & Agent Systems

## Assignment 5

Deadline: October 16th 22:00

### Report requirements

- Hand in your updated Python code and a PDF report with your answers to the questions below.
- Include your student numbers and names in the code and the report itself.
- If you have to write code, add the small snippets of code to the report. You can use the *listings* package for this.<sup>1</sup>

### Part 1:

#### Question 1

Consider the argumentation framework in Figure 1 and answer the following questions:

- Is the set  $\{A, B, F\}$  admissible? Why (not)?
- Is the set  $\{C, E, F\}$  admissible? Why (not)?
- Is the set  $\{A, E\}$  admissible? Why (not)?
- Explain why it is not possible to have an admissible set containing  $D$ .

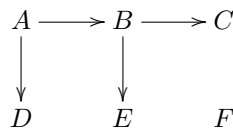


Figure 1

---

<sup>1</sup>The *listings* package is explained at [https://en.wikibooks.org/wiki/LaTeX/Source\\_Code\\_Listings](https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings).

## Question 2

Imagine the following situation:

You go to the park or the beach or the cinema. It is too windy to go to the beach. It is too rainy to do something outside without an umbrella. However, you take an umbrella.

A model of this situation in terms of arguments and attack relations is shown in Figure 2. Use the abbreviations (Cinema, Park, Umbrella, Windy, Beach, Rainy) to refer to the arguments.

- What are the admissible sets of arguments?
- What are the complete extensions of arguments?
- What are the preferred extensions of arguments?
- What are the stable extensions of arguments?

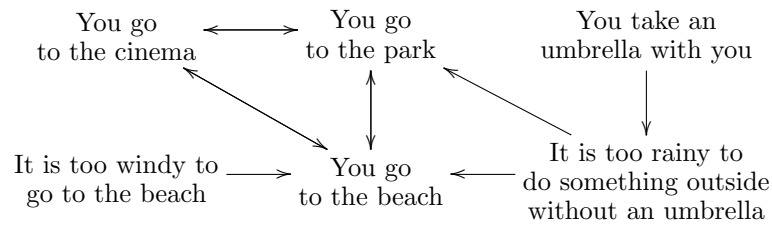


Figure 2

## Question 3

Consider the argumentation framework in Figure 3.

- Provide 4 different admissible sets.
- Is there an admissible set containing  $B$ ? Explain your answer.
- Provide all stable extensions.

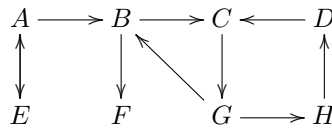


Figure 3

## Part 2: Programming

Download the code folder. In it you will find an incomplete Python program for finding the conflict-free sets, admissible sets, complete extensions and stable extensions of an argumentation framework. The program works by gradually labeling the arguments either IN or OUT. Once this is done it checks if the arguments labeled IN qualify as a set of the desired extension.

Input files must be provided in trivial graph format.<sup>2</sup> In that format, first, the argument names are listed followed by a '#' symbol. Then, the attack relations between arguments are listed, with first the attacking and then the attacked argument.

So far, the program can find conflict-free sets (e.g., `python solver.py conflict-free frame1.tgf`). Note that it can take a long time (e.g., for `frame3.tgf`).

1. Complete the program such that it is also able to find admissible sets, complete extensions and stable extensions.
2. Testing out every possible partition into IN and OUT arguments is very inefficient. Adapt the function `find_extensions()` to increase performance. The resulting code should be able to solve `frame3.tgf` within a reasonably short time.

---

<sup>2</sup>See [https://en.wikipedia.org/wiki/Trivial\\_Graph\\_Format](https://en.wikipedia.org/wiki/Trivial_Graph_Format).