

# Computing the extensions of Abstract Argumentation Frameworks

**Exercise 1:** Give all complete, grounded, preferred and stable extensions of the argumentation frameworks depicted in Figures 1-4.

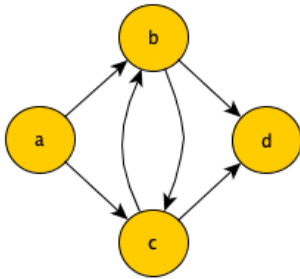


Figure 2

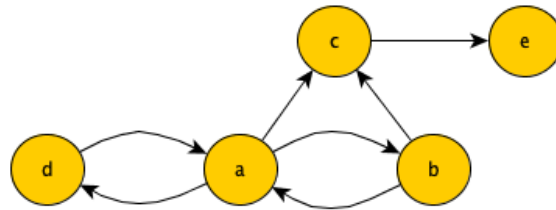


Figure 1

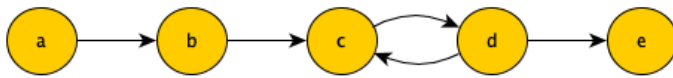


Figure 3

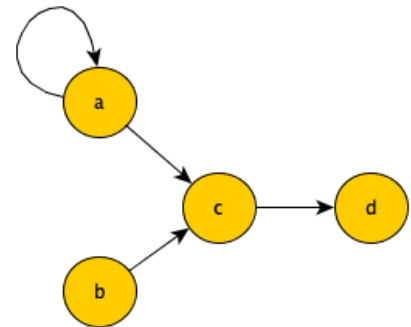


Figure 4

**Exercise 2:** Verify your answers to exercise 1 using an ASP implementation of AAFs.

1. Download the .dl files in your computer. Each of the .dl files implements in ASP a different acceptability semantics of AAFs.
2. Encode each of the frameworks shown in Figures 1-4 in ASP (as shown below) and save each of them in a separate file in your computer:
  - Encode each argument (e.g. a) as a statement of the form: `arg(a)`.
  - Encode each attack (e.g. from a to b) as a statement of the form: `att(a,b)`.
3. Open Clingo at: <https://potassco.org/clingo/run/> and copy the following code into the text box:
  - a. The ASP code of the semantics you want to use from the corresponding .dl file (from step 1)
  - b. The ASP encoding of the example argumentation framework you want to compute the extensions of (from the corresponding file you created in step 2)
  - c. Add the following clause at the end: `#show in/1.`  
This instructs Clingo to show only the 'in' atoms (representing the arguments contained in each extension).
4. From the drop-down list next to 'reasoning mode' select 'Enumerate all'.
5. Click the Run! button: Each answer set corresponds to a different extension of the example AAF under the selected semantics.

**Exercise 3:** A naive extension is defined as a maximal (w.r.t. set inclusion) conflict-free set. Can you write an ASP program, with which we can compute the naive extensions of an AAF?