Instantiation-based Argumentation

A prominent approach to formal argumentation is instantiation-based argumentation:
1. start from a knowledge base (KB), which is potentially inconsistent;
2. from KB, all relevant arguments are constructed;
3. relationship between arguments is analyzed;
4. abstract away from the contents of the arguments and only consider the remaining abstract argumentation framework (AF);
5. semantics for AFs deliver a collection of sets of arguments (‘extensions’) which are understood as jointly acceptable;
6. re-interpret extensions in terms of their claims.

Claim-centric Reasoning Problems

Given semantics σ, a CAF CF = (A,R,claim), claim c ∈ C, and claims C ⊆ C.
• CredCAF: Does c ∈ C for at least one S ∈ σ(CF)?
• SkeptCAF: Does c ∈ C for all S ∈ σ(CF)?
• VerCAF: Does C ∈ σ(CF) hold?
• NEmptyCAF: Does S ∅ hold for some S ∈ σ(CF)?


Reasoning Modes

• Argument-centric Reasoning: is a particular argument accepted w.r.t. the extensions?
• Claim-centric Reasoning: is a particular claim accepted w.r.t. the extensions?

Skeptical Acceptance: is a particular argument accepted w.r.t. the extensions?

Example: Instantiating AFs from Logic Programs

Consider the following logic program:
\[ P = \{ \alpha \leftarrow \alpha \land \neg \beta, \beta \leftarrow \alpha \land \neg \beta, \gamma \leftarrow \alpha \land \neg \beta \} \]

The instantiation yields an AF FP = (A,R) with arguments \( A = \{ \alpha, \beta, \gamma \} \), where
• \( \alpha \) represents rule r1 and has claim a;
• \( \beta \) represents rule r2 with claim b;
• \( \gamma \) and \( \beta \) represent rules r3 and r4 respectively, both have as claim c.

An argument representing rule r attacks an argument representing rule r’ if the head of r occurs negated in the rule body of r’. Hence, R = \( \{ (\alpha, \beta, \gamma, \alpha, \beta), (\alpha, \beta, \gamma, \beta, \gamma) \} \).

Stable model semantics of logic programs corresponds to stable extensions of AFs:
• the two stable models \( S_1 = \{ a, c \} \) and \( S_2 = \{ b, c \} \) of P are given via
• the two stable extensions \( E_1 = \{ \alpha, \beta, \gamma \} \) and \( E_2 = \{ \beta, \gamma \} \) of FP;
• the claims of \( E_1 \) yield \( S_1 \) and those of \( E_2 \) yield \( S_2 \).

Observation:
• Argument acceptance alone is insufficient to decide the acceptance of claims.

Introducing the Tractability Frontier

We follow three directions towards tractability results:

Exploiting Special Graph Classes

Some results are in contrast to argument-centric reasoning:
• SkeptCAF naive, SkeptCAF naive, VerCAF naive, VerCAF naive remain coNP-hard for acyclic CAFs.
• For \( \sigma \in \{ \text{naive}, \text{stb}, \text{adm} \} \), SkeptCAF naive is coNP-complete for bipartite well-formed CAFs.

Exploiting the Number of Claims

We parameterize the problems with the number k of different claims that appear in the CAF, and obtain a Fixed-Parameter Tractability Result:
• CredCAF, SkeptCAF, and VerCAF can be solved in time \( O(k^2 \cdot poly(n)) \) for \( \sigma \in \{ \text{naive}, \text{stb}, \text{adm}, \text{com}, \text{pref} \} \).

Exploiting (Incidence) Tree-Width of CAFs

We introduce the parameter incidence tree-width of well-formed CAFs which measures the structure of the interplay between claims and arguments and is complementary to tree-width.

Main Results (for \( \sigma \in \{ \text{naive}, \text{stb}, \text{adm}, \text{com}, \text{pref} \} \)):
• CredCAF and SkeptCAF are fixed-parameter tractable w.r.t. the tree-width;
• VerCAF in NP-hard for CAFs of tree-width 1;
• CredCAF, SkeptCAF, and VerCAF are fixed-parameter tractable w.r.t. incidence tree-width.

Main References