The KWT Benchmark Generator for ICCMA 2025

Isabelle Kuhlmann
Artificial Intelligence Group
University of Hagen
Germany
isabelle.kuhlmann@fernuni-hagen.de

Matthias Thimm
Artificial Intelligence Group
University of Hagen
Germany
matthias.thimm@fernuni-hagen.de

***Abstract*—We present a generator for abstract argumentation frameworks that produces instances which are particularly challenging for the task of deciding skeptical acceptability w.r.t. preferred semantics.**

I. Introduction

Over the past years, the International Competition on Computational Models of Argumentation (ICCMA)[1] has significantly contributed to the evaluation of argumentation systems. Nevertheless, the process of generating benchmarks for different argumentation formalisms remains a perpetual task that possesses numerous challenges (see [1] for a discussion).

With the *KWT Benchmark Generator*, we focus on generating benchmarks for problems related to abstract argumentation frameworks [2]; more precisely, our aim is to provide challenging benchmarks for the task of deciding skeptical acceptability w.r.t. preferred semantics. Although this task is generally Π_2^P-complete [3], in practice it can often be solved much more efficiently by exploiting some "shorcuts". For instance, each argument in the grounded extension (which can be computed in polynomial time) and each argument in the ideal extension (whose corresponding problems are Θ_2^P-complete) is also skeptically accepted w.r.t. preferred semantics. W.r.t. a set of ICCMA'17 benchmarks, it has been pointed out that a majority of arguments that are skeptically accepted under preferred semantics is also included in the grounded extension [4]. Thus, in the majority of cases, it is sufficient to solve a less complex problem, which may distort the interpretation of experimental results. To address this problem, we propose the use of the *KWT Benchmark Generator*[2], which is designed to circumvent the aforementioned problem.

II. Preliminaries

An (abstract) *argumentation framework* (AF) [2] is a pair $F = (\mathsf{Arg}, R)$, with Arg being a set of arguments and $R \subseteq \mathsf{Arg} \times \mathsf{Arg}$ a relation between those arguments. An argument $a \in \mathsf{Arg}$ *attacks* an argument $b \in \mathsf{Arg}$ if $(a, b) \in R$. Moreover, we define the set of arguments attacking a given argument a as $a_F^- = \{b \mid (b, a) \in R\}$, and the set of arguments being attacked by a as $a_F^+ = \{b \mid (a, b) \in R\}$. In the same fashion we define E_F^- and E_F^+ for a set $E \subseteq \mathsf{Arg}$. We call an argument $a \in \mathsf{Arg}$ *defended* by a set of arguments $E \subseteq \mathsf{Arg}$ if every argument $b \in \mathsf{Arg}$ that attacks a is itself attacked by some argument $c \in E$, i.e., if $a_F^- \subseteq E_F^+$.

Further, a set $E \subseteq \mathsf{Arg}$ is *conflict-free* if $E \cap E_F^+ = \emptyset$. If a set $E \subseteq \mathsf{Arg}$ is conflict-free and each $a \in E$ is defended by E, we call E admissible (ad). We call sets of jointly acceptable arguments *extensions*, which can be defined under various semantics. The classical semantics, following the seminal work by Dung [2], are defined as follows:

- A set $E \subseteq \mathsf{Arg}$ is *complete* (co) iff it is admissible, and if E defends $a \in \mathsf{Arg}$ then $a \in E$.
- A set $E \subseteq \mathsf{Arg}$ is *grounded* (gr) iff E is complete and $\subseteq$-minimal.
- A set $E \subseteq \mathsf{Arg}$ is *preferred* (pr) iff E is complete and $\subseteq$-maximal.
- A set $E \subseteq \mathsf{Arg}$ is *stable* (st) iff it is complete and $E \cup E_F^+ = \mathsf{Arg}$.

In addition, we define a set $E \in \mathsf{Arg}$ to be an *ideal* (id) extension [5] if E is admissible, for every preferred extension E', it holds that $E \subseteq E'$, and E is $\subseteq$-maximal with these two properties. Note that the grounded and the ideal extension of an AF are each a uniquely defined, and that the former is always a subset of the latter.

III. The KWT Benchmark Generator

The main idea behind the *KWT Benchmark Generator* is to avoid (to a high degree) producing argumentation frameworks that are "easy" to solve, i. e., instances in which arguments that are skeptically accepted under preferred semantics are also in the grounded or the ideal extension. Another "easy" case regarding this task occurs when arguments are attacked by some admissible set—such arguments are never skeptically accepted w.r.t. pr—and deciding this is a problem in NP. The generator takes the parameters

- num_{args}: the total number of arguments,
- num_{pa}: the number of arguments to be skeptically accepted under preferred semantics,
- num_{cred}: the number of arguments to be contained in at least one preferred extension,
- num_{pref}: the number of preferred extensions,
- num_{ideal}: the number of arguments in the ideal extension,

in addition to 7 further parameters that control the probability of attacks between different sets of arguments. More precisely,

[1] http://argumentationcompetition.org/index.html

[2] Note that this generator was first used in [4] and later described in more detail in [4].

these parameters set the probabilities of arguments in the ideal extension to be attacked and to attack back, respectively, the probabilities of credulously accepted arguments to be attacked and to attack back, the probabilities of skeptically accepted arguments that are not contained in the ideal extension to be attacked and to attack back, and the probability of further random attacks between unaccepted arguments. Given these parameters, a random AF F is generated as follows:

1) The set Arg of num_{args} arguments is created and arguments are associated to sets S_{pa} (skeptically accepted arguments w.r.t. preferred semantics), S_{ideal} (arguments in the ideal extension), S_{cred} (arguments that are credulously accepted w.r.t. preferred semantics), S_{unacc} (arguments that are not credulously accepted w.r.t. preferred semantics), such that $S_{ideal} \subseteq S_{pa} \subseteq S_{cred}$, $S_{cred} \cup S_{unacc} = \mathsf{Arg}$, and the corresponding cardinalities are respected. Finally, sets $E_1, \ldots, E_{num_{pref}}$ (the preferred extensions) are created by adding all arguments from S_{pa} and randomly drawn arguments from $S_{cred} \setminus S_{pa}$.
2) For every argument $a \in S_{ideal}$, random attackers from S_{unacc} are sampled. For each of these attackers b, another argument from S_{ideal} is sampled that attacks b. This ensures that the grounded extension will be empty and that the ideal extension is capable of defending itself (thus forming an admissible set).
3) For every argument $a \in S_{pa} \setminus S_{ideal}$, attacks from unaccepted arguments are sampled in a similar way (to ensure an empty grounded extension). Furthermore, every such argument a must be defended by each preferred extension. Thus, for each preferred extension E, some arguments are sampled to defend a.
4) For every preferred extension E and $a \in E \setminus S_{pa}$, attackers for a are sampled from $\mathsf{Arg} \setminus E$ and corresponding defenders are defined within E.
5) Additional random attacks are added between arguments in S_{unacc}.
6) In order to avoid having stable extensions (which may also ease computation of arguments that are skeptically accepted under preferred semantics, since every stable extension is also preferred), we add another self-attacking argument and some attacks between this argument and arguments from S_{unacc}.

Note that due to the random approach of generating an argumentation graph, it may not necessarily be the case that the number of skeptically/credulously accepted arguments (w.r.t. preferred semantics) as well as the number of arguments in the ideal extension exactly match the given parameters. However, our experiments in [4] showed that it is indeed relatively hard to decide skeptical acceptance (w.r.t. preferred semantics) for most arguments in the resulting graph.

The graph generator[3] and an example demonstrating its usage[4] we used can be found online.

[3]http://tweetyproject.org/r/?r=kwt_gen

[4]http://tweetyproject.org/r/?r=kwt_gen_ex

IV. Conclusion

We described the *KWT Benchmark Generator*, which aims at producing challenging argumentation frameworks to evaluate the task of deciding skeptical acceptance under preferred semantics by largely avoiding the production of instances that can be solved by reverting to a less complex problem.

References

[1] I. Kuhlmann and M. Thimm, "A discussion of challenges in benchmark generation for abstract argumentation." in *Arg&App@ KR*, 2023, pp. 78–84.

[2] P. M. Dung, "On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games," *Artificial Intelligence*, vol. 77, no. 2, pp. 321–358, 1995.

[3] P. E. Dunne and T. J. M. Bench-Capon, "Coherence in finite argument systems," *Artificial Intelligence*, vol. 141, no. 1/2, pp. 187–203, 2002.

[4] I. Kuhlmann, T. Wujek, and M. Thimm, "On the impact of data selection when applying machine learning in abstract argumentation," in *Computational Models of Argument*. IOS Press, 2022, pp. 224–235.

[5] P. M. Dung, P. Mancarella, and F. Toni, "Computing ideal sceptical argumentation," *Artificial Intelligence*, vol. 171, no. 10-15, pp. 642–674, 2007.