Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200496

Timed Abstract Dialectical Frameworks: A Simple Translation-Based Approach

Ringo BAUMANN [a], Maximilian HEINRICH [a]
[a] *{baumann, mheinrich}@informatik.uni-leipzig.de*

Abstract. Abstract dialectical frameworks (ADFs) are one of the most powerful generalization of classical Dung-style AFs. In this paper we show how to use ADFs if we want to deal with acceptance conditions changing over time. We therefore introduce so-called *timed abstract dialectical frameworks (tADFs)* which are essentially ADFs equipped with time states. Beside a precise formal definition of tADFs and an illustrating example we prove that Kleene's three-valued logic $\mathcal{K}_3$ facilitate the evaluation of acceptance functions if we do not allow multiple occurrences of atoms.

Keywords. Abstract Dialectical Frameworks, Time, Three-valued Logics

Introduction

Argumentation has become one of the major fields within AI over the last two decades [1,2]. In particular, Dung's abstract argumentation frameworks (AFs) are a by now widely used formalism [3]. Main reasons for this success story are the simplicity of AFs and the plethora of existing semantics [4], the ability to reconstruct mainstream nonmonotonic formalisms [3] as well as their potential to be used as core method in advanced argumentation formalisms [5,6]. However, through the years the community realized that the limited expressive capability of AFs, namely the option of single attacks only, reduce their suitability as right target systems for more complex applications [7]. Therefore a number of additional functionality were introduced encompassing preferences, values, collective attacks, attacks on attacks as well as support relations between arguments [8,9,10,11,12]. One of the most powerful generalizations of Dung AFs, yet staying on the abstract layer, are so-called *abstract dialectical frameworks* (ADFs) [13]. The additional expressive power is achieved by adding acceptance conditions to the arguments which allow for the specification of arbitrary relationships between arguments and their parents in the argument graph.

In this paper we show how to use classical ADFs if we are faced with conditions changing over time. We therefore introduce so-called *timed abstract dialectical frameworks* (tADFs) which are essentially classical ADFs plus time states. In this way we are able to speak about the same statement s at different time points t. For instance, an acceptance condition like $\phi_{s_4} = a_1 \vee a_2 \vee a_3$ encodes that s should be accepted at time point 4 if statement a is at least ones accepted between time points 1 and 3. If the numbers are interpreted as the first months of the year and if s and a are standing for "I am on vacation in France" or "I have a salary increase", respectively, then ϕ_{s_4} expresses "I will be vacationing in France in April, if I get a salary increase between January and March."

The paper is organized as follows: Section 1 reviews necessary background regarding ADFs. In Section 2 we proceed with the formal introduction of tADFs and a presentation of useful timed acceptance conditions. Moreover, we give an illustrating example. Section 3 provides two theoretical insights regarding the evaluation of acceptance functions with the help of three-valued logics. Finally, Section 4 discusses related work and give pointers for future work.

1. Background

1.1. Classical ADFs, Information Order and Consensus

The definition of ADFs [14] was motivated by the effort to obtain more expressive power than classical AFs. This is achieved by equipping each argument with a so-called acceptance condition which can be given as a logical formula [15].

Definition 1. *An* abstract dialectical framework *is a tuple $D = (S, \Phi)$ where S is a set of statements and $\Phi = \{\varphi_s \mid s \in S\}$ is a set of propositional formulae.*

The formal definitions of the different semantics are based on three-valued operators which handle two-valued interpretations.

Definition 2. *Let $D = (S, \Phi)$ be an* ADF. *A two-valued resp. three-valued interpretation v for D is a total function $v : S \mapsto \{\mathbf{t}, \mathbf{f}\}$ or $v : S \mapsto \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. We use $\mathcal{V}_2^D$ and $\mathcal{V}_3^D$ for the set of all two resp. three valued interpretations for D.*

Next we define the so-called *information order*. It orders the three values $\mathbf{u}$ (undecided), $\mathbf{t}$ (true) and $\mathbf{f}$ (false) based on their information content.

Definition 3. *Let $D = (S, \Phi)$ be an ADF. The* information order *$\leq_i$ over $\{\mathbf{t},\mathbf{f},\mathbf{u}\}$ is the reflexive closure of $<_i$, where $\mathbf{u} <_i \mathbf{t}$ and $\mathbf{u} <_i \mathbf{f}$. This is generalised for three-valued interpretations for D in a point-wise fashion:*

$$v_1 \leq_i v_2 \textit{ if and only if } \forall s \in S : v_1(s) \in \{\mathbf{t}, \mathbf{f}\} \implies v_1(s) = v_2(s).$$

The consensus operator *$\sqcap_i$ assigns $\mathbf{t} \sqcap_i \mathbf{t} = \mathbf{t}$, $\mathbf{f} \sqcap_i \mathbf{f} = \mathbf{f}$, and $\mathbf{u}$ otherwise.*

Let $\mathtt{u} \in \mathcal{V}_3^D$, s.t. $\mathtt{u}(s) = \mathbf{u}$ for any $s \in S$. Note that for any $v \in \mathcal{V}_3^D, \mathtt{u} \leq_i v$. This means, $\mathtt{u}$ is the $\leq_i$-least element in $\mathcal{V}_3^D$. We will call $\mathtt{u}$ the *least information interpretation*. Moreover, for $v \in \mathcal{V}_3^D$ we define $[v]_2^D = \{w \in \mathcal{V}_2^D \mid v \leq_i w\}$. This means, $[v]_2^D$ contains all two-valued completions of v.

1.2. Semantics

To define the semantics the approximation fixpoint theory of Denecker, Marek, and Truszczyński [16] has been used.

Definition 4. *Given an ADF $D = (S, \Phi)$. We define $\Gamma_D : \mathcal{V}_3^D \mapsto \mathcal{V}_3^D$ as*

$$\Gamma_D(v) : S \mapsto \{\mathbf{t}, \mathbf{f}, \mathbf{u}\} \textit{ with } s \mapsto \sqcap_i\{w(\varphi_s) \mid w \in [v]_2^D\}.$$

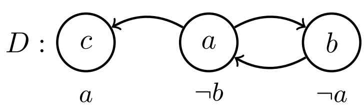
The idea behind the operator is, that based on a given three-valued interpretation, it is checked for every two-valued interpretation with at least as much information whether a consensus on the valuation of the acceptance conditions can be found. If all two valued interpretations consent on either $\mathbf{t}$ or $\mathbf{f}$, then the respective truth value can be assigned by the operator, otherwise it will be evaluated with $\mathbf{u}$. In the following we introduce so-called admissible, complete, preferred and grounded interpretation (abbr. by *adm*, *cmp*, *prf*, *grd*).

Definition 5. *Given an ADF $D = (S, \Phi)$ and $v \in \mathcal{V}_3^D$.*

1. $v \in adm(D)$ *if and only if* $v \leq_i \Gamma_D(v)$,
2. $v \in cmp(D)$ *if and only if* $v = \Gamma_D(v)$,
3. $v \in prf(D)$ *if and only if v is $\leq_i$-maximal in* $cmp(D)$,
4. $v \in grd(D)$ *if and only if v is $\leq_i$-least in* $cmp(D)$.

The definitions above justify the following two subset chains for any ADF D, namely $prf(D) \subseteq cmp(D) \subseteq adm(D)$ as well as $grd(D) \subseteq cmp(D) \subseteq adm(D)$.

Example 1. *Consider the ADF $D = (\{a, b, c\}, \{\phi_a = \neg b, \phi_b = \neg a, \phi_c = a\})$. Let*



us verify that $\{\mathbf{u}\} = grd(D)$. It suffices to show that $\mathbf{u}$ satisfies $\mathbf{u} = \Gamma_D(\mathbf{u})$. Note that $\leq_i$-leastness is immediately apparent since $\mathbf{u}$ is even $\leq_i$-least in $\mathcal{V}_3^D$. Consider the two-valued interpretation I_1, I_2, s.t. $I_1(a) = I_1(b) = I_1(c) = \mathbf{t}$ and $I_2(a) = I_2(b) = I_2(c) = \mathbf{f}$. We obtain $I_1(\phi_a) \sqcap_i I_2(\phi_a) = \mathbf{u}$ since $I_1(\phi_a) = I_1(\neg b) = \mathbf{f}$ and $I_2(\phi_a) = I_2(\neg b) = \mathbf{t}$. Analogously, one may easily check that $I_1(\phi_b) \sqcap_i I_2(\phi_b) = \mathbf{u}$ and $I_1(\phi_c) \sqcap_i I_2(\phi_c) = \mathbf{u}$ justifying $\mathbf{u} = \Gamma_D(\mathbf{u})$. The other semantics are given as $adm(D) = \{v_1, v_2, v_3, v_4, \mathbf{u}\}$, $cmp(D) = \{v_1, v_3, \mathbf{u}\}$, $prf(D) = \{v_1, v_3\}$ with $v_1 = \{a : \mathbf{t}, b : \mathbf{f}, c : \mathbf{t}\}$, $v_2 = \{a : \mathbf{t}, b : \mathbf{f}, c : \mathbf{u}\}$, $v_3 = \{a : \mathbf{f}, b : \mathbf{t}, c : \mathbf{f}\}$ and $v_4 = \{a : \mathbf{f}, b : \mathbf{t}, c : \mathbf{u}\}$.

2. Temporal Aspects and Timed ADFs

2.1. Timed Abstract Dialectical Framework

The classical definition of ADFs does not provide one with temporal notions. However, in daily life we are often faced with statements/laws which are valid for a certain time only or depend on the past development, e.g. "You can continue working in the company as long as the Brexit is not delivered." or "From the beginning of next year it will be not allowed to build a nightclub near a residential area.". In order to encode statements like the ones before we need to be able to distinguish between different time states related via a certain ordering. In this very first paper we decided to keep things as simple as possible. Nevertheless, we will see that this approach is powerful enough to model many frequently occuring temporal restrictions. More precisely, a *timed abstract dialectical framework* (tADF) is a classical ADF equipped with a countable set T of time states. We assume that this set is totally ordered, i.e. there is a binary relation $\leq$ over T which is antisymmetric, transitive and connex. Many times T will simply be a subset of the first natural numbers with the inherited standard ordering. Hereby, a certain time state n might stand for an hour, a day, a week or a month or whatever granularity is needed. In this way we are able to speak about the same statement s at different time points t in the future, denoted as s_t. Accordingly, we will have timed acceptance conditions ϕ_{s_t} for any statement s at any time point t.

Definition 6. *A* timed abstract dialectical framework *(for short,* tADF*) is a tuple $D = (S, T, \Phi)$ where S is a set of statements, T total ordered set of time states and $\Phi = \{\varphi_{s_t} \mid s \in S, t \in T\}$ is a set of propositional formulae, one for each statement $s \in S$ and time state $t \in T$.*

In tADFs we treat each argument at each time step as one single classical statement. This means, a tADF with n statements and m time states corresponds to a classical ADF with $n \cdot m$ statements. Moreover, the definition of tADFs allows us to apply the standard semantics of classical ADFs (cf. Example 2).

2.2. Temporal Acceptance Functions

To facilitate the use of tADFs we introduce additional temporal shorthands, which can be used for the corresponding acceptance conditions. Note that any shorthand can be retranslated to classical propositional logic. Given $D = (S, T, \Phi)$ and statements $a, c \in S$ as well as a time interval $[i, j] \subseteq T$.

1. $\varphi_{c_t} = a^{[i,j]}_{\geq 1} := \bigvee_{i \leq k \leq j} a_k$.
   This formula expresses that c should be accepted at time state t, if a is **at least ones accepted** in $[i, j]$. Hence, a supports c at least ones inbetween time states i and j.
2. $\varphi_{c_t} = a^{[i,j]}_{\geq n} := \bigvee_{\substack{\{k_1,\ldots,k_n\} \subseteq [i,j] \\ |\{k_1,\ldots,k_n\}| = n}} a_{k_1} \wedge \ldots \wedge a_{k_n}$.
   This formula expresses that c should be accepted at time state t, if a is **at least n-times accepted** in $[i, j]$. This means, a supports c at least n-times during the time interval $[i, j]$.
3. $\varphi_{c_t} = a^{[i,j]}_{\leq n} := \neg(a^{[i,j]}_{\geq n+1}) = \bigwedge_{\substack{\{k_1,\ldots,k_{n+1}\} \subseteq [i,j] \\ |\{k_1,\ldots,k_{n+1}\}| = n+1}} \neg a_{k_1} \vee \ldots \vee \neg a_{k_{n+1}}$.
   This formula expresses that c should be accepted at time state t, if a is **at most n-times accepted** in $[i, j]$. This means, an n-fold acceptance of a during the time interval $[i, j]$ prevents the acceptance of c.
4. $\varphi_{c_t} = a^{[i,j]}_{\leq 1} := \neg(a^{[i,j]}_{\geq 2}) = \bigwedge_{\substack{\{k_1,k_2\} \subseteq [i,j] \\ |\{k_1,k_2\}| = 2}} \neg a_{k_1} \vee \neg a_{k_2}$.
   For the sake of completeness we also present an important instantiation of the timed acceptance formula above, namely $a^{[i,j]}_{\leq 1}$ expressing that c should be accepted at time state t, if a is **at most ones accepted** in $[i, j]$.
5. $\varphi_{c_t} = a^{[i,j]}_{=n} := \varphi_{c_t} = a^{[i,j]}_{\leq n} \wedge a^{[i,j]}_{\geq n}$
   This formula expresses that c should be accepted at time state t, if a is **exactly n-times accepted** in $[i, j]$.

A timed ADF as well as the above introduced shorthands are illustrated in the following example.

Example 2. *Suppose that Charles is making plans for the first months of the new year. He will spend his vacation (v) in France in April if he gets a salary raise (s) in the months before the vacation. In order to get to the desired location he would like to take a plane (p). Unfortunately, such a flight line (l) is currently only planned but Charles knows that it will be introduced between March and May. If no flight is available, he will take the train (t).*

This example can be therefore represented as a tADF $D = (S, T, \Phi)$ *where* $S = \{v, s, t, p, l\}$ *and* $T = \{1, 2, 3, 4, 5\}$ *(cf. Figure 1). Here, any time state* $n \in T$ *corresponds to the* n^{th} *month of the year as expected. The acceptance functions are listed in Table 2. For instance, the formula* φ_{l_4} *expresses that the flight line will be set up in April, if it is neither introduced in March, nor in May.* φ_{l_4} *supports vacation in France provided that Charles received at least one raise in the first three months of the year and if a train or plane goes there. Moreover, the condition* φ_{t_4} *encodes that Charles will take the train if there is no plane available in April and finally,* φ_{p_4} *expresses that Charles will take an airplane if the flight connection has been established previously and if he is not traveling by train. Salary increases are possible for any month and do not depend on other events. Consequently,* $\varphi_{s_i} = s_i$ *for any* $i \in \{1, 2, 3, 4, 5\}$*.*

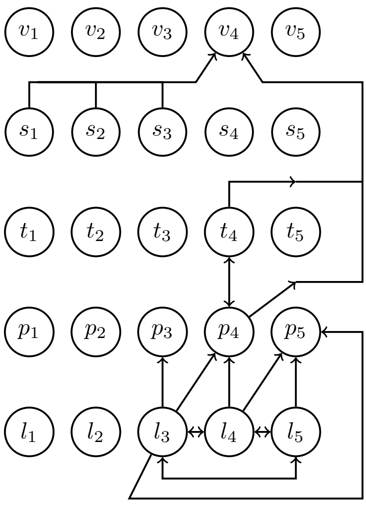


Figure 1. The tADF D

a_t	φ_{a_t}
v_1, v_2, v_3, v_5	$\bot$
v_4	$s_{\geq 1}^{[1,3]} \wedge (p_4 \vee t_4)$
t_1, t_2, t_3, t_5	$\top$
t_4	$\top \wedge \neg p_4 \equiv \neg p_4$
p_1, p_2	$\bot$
p_3	l_3
p_4	$l_{\geq 1}^{[3,4]} \wedge \neg t_4$
p_5	$l_{\geq 1}^{[3,5]}$
l_1, l_2	$\bot$
l_3	$\neg \left(l_{\geq 1}^{[4,5]} \right)$
l_4	$\neg l_3 \wedge \neg l_5$
l_5	$\neg \left(l_{\geq 1}^{[3,4]} \right)$
s_i	s_i

Table 1. Acceptance functions of D

For the evaluation of the tADFs D we use classical ADF semantics. In the following we stick to preferred interpretations as they maximize the information content which appears desirable for the planning context. Table 2 shows 8 out of forty preferred interpretations[1] *of the tADF D. Any interpretation describes a possible scenario. The selected interpretations agree on the availability of the plane in May since for any considered scenario the flight line was only introduced in*

[1] All preferred interpretation can be found under `https://github.com/kmax-tech/ADF`.

May meaning that Charles has to take the train in April in order to get to France. The first interpretation v_1 expresses that the vacation cannot take place since no salary increase happened in the months before. In any other interpretations one or more salary increases happened implying that Charles can take his vacation.

$prf(D)$	l_3	l_4	l_5	p_3	p_4	p_5	s_1	s_2	s_3	t_4	v_4
v_1	**f**	**f**	**t**	**f**	**f**	**t**	**f**	**f**	**f**	**t**	**f**
v_2	**f**	**f**	**t**	**f**	**f**	**t**	**f**	**f**	**t**	**t**	**t**
v_3	**f**	**f**	**t**	**f**	**f**	**t**	**f**	**t**	**f**	**t**	**t**
v_4	**f**	**f**	**t**	**f**	**f**	**t**	**f**	**t**	**t**	**t**	**t**
v_5	**f**	**f**	**t**	**f**	**f**	**t**	**t**	**f**	**f**	**t**	**t**
v_6	**f**	**f**	**t**	**f**	**f**	**t**	**t**	**f**	**t**	**t**	**t**
v_7	**f**	**f**	**t**	**f**	**f**	**t**	**t**	**t**	**f**	**t**	**t**
v_8	**f**	**f**	**t**	**f**	**f**	**t**	**t**	**t**	**t**	**t**	**t**

Table 2. Selected preferred interpretations of D.

3. Evaluation of Acceptance Functions and Three-Valued Logics

In order to facilitate the use of (t)ADFs, we developed a Python script[2], which enables an easy calculation of the desired semantics. During creation of the script the questions occurred, whether the computational expensive calculation of the gamma operator can be somehow simplified. According to Definition 4 the operator takes a three-valued interpretation v and outputs a three-valued one v'. More precisely, for any statement s we have to evaluate the corresponding acceptance function φ_s w.r.t. all two-valued completions of v. Now, applying the consensus operator on these two-valued outputs leaves us with the assignment to s under v'. The idea was to use a three-valued logic $\mathcal{L}_3$, s.t. the evaluation of φ_s can be done directly in $\mathcal{L}_3$ without any computation of two-valued completions and the use of the consensus operator. The following theorem shows that this endeavour is doomed to failure.

Theorem 1. *There is no truth-functional three-valued logic $\mathcal{L}_3$, s.t. for any propositional formula φ and any three-valued interpretation v:*

$$v^{\mathcal{L}_3}(\varphi) = \sqcap_i \{w(\varphi) \mid w \in [v]_2\}.$$

The decisive point for the impossibility of using a three-valued logic in general is that two-valued completions of parts of a composed formula cannot be considered independently. However, such behaviour can be enforced if considering acceptance conditions where each atom appears at most ones. We therefore define the following fragment of classical propositional logic. Let $\mathcal{A} = \{a, b, c, ...\}$ be the set of atomic formulas and $\sigma(\varphi)$ the set of all atoms occuring in φ, e.g. for $\varphi = a \vee \neg a$ we have $\sigma(\varphi) = \{a\}$.

Definition 7. *The set $\mathcal{F}$ is defined inductively as:*

1. $\mathcal{A} \subseteq \mathcal{F}$,
2. *If* $\varphi \in \mathcal{F}$, *then* $\neg\varphi \in \mathcal{F}$,
3. *If* $\varphi, \psi \in \mathcal{F}$ *and* $\sigma(\varphi) \cap \sigma(\psi) = \emptyset$, *then* $\varphi \vee \psi, \varphi \wedge \psi \in \mathcal{F}$.

[2]Submitted to SAFA 2020. `http://safa2020.argumentationcompetition.org/`

a	b	$a \vee b$	$a \wedge b$	$\neg a$
t	**t**	**t**	**t**	**f**
t	**f**	**t**	**f**	**f**
t	**u**	**t**	**u**	**f**
f	**t**	**t**	**f**	**t**
f	**f**	**f**	**f**	**t**
f	**u**	**u**	**f**	**t**
u	**t**	**t**	**u**	**u**
u	**f**	**u**	**f**	**u**
u	**u**	**u**	**u**	**u**

Table 3. Kleene's three-valued logic $\mathcal{K}_3$

It is easy to see that any formula $\varphi \in \mathcal{F}$ does not have multiple occurrences of atoms. The following theorem shows that if restricting acceptance functions to $\mathcal{F}$ the use of Kleenes strong three valued logic $\mathcal{K}_3$ [17] is enabled. The thruth tables regarding disjunction, conjunction and negation are given in Table 3.

Theorem 2. *For any $\varphi \in \mathcal{F}$ and any three-valued interpretation v we have:*

$$v^{\mathcal{K}_3}(\varphi) = \sqcap_i \{w(\varphi) \mid w \in [v]_2\}.$$

4. Discussion and Conclusion

The concept of time in regard to argumentation is not new. In [18] a timed argumentation framework (TAFs) is considered, which can be used for classical AFs and bipolar AFs [12]. In comparison tADFs are offering a more fine-grained approach, because not only pure attack and support relations between nodes can be considered but also mixed forms. In addition tADFs are offering the possibility to make statements about events which depends on other timesteps in the past or the near future. Therefore it is not required to consider a specific time-interval as in TAFs. An other approach to the time topic is the LARS-framework [19] which uses a logic-based framework and a window operator for modeling datatstreams at given time-intervals. Here the focus is on a continous stream of input and evaluation of possible actions. Timed ADFs are designed to consider all time points through the defined acceptance conditions. Therefore there is no narrowing to a current time step with information available at that moment, through this could be considered with specifc semantics. The definition of tADFs allows us to use all theoretical results about ADFs. In order to facilitate the calculation of ADFs semantics, we introduced a special subclass of formulas, where the value of the gammaoperator can be calculated directly with Kleenes strong-three valued logic. Also it could be shown that no three-valued logic in general can exist in order to model the gammaoperator. In further research we want to evaluate, whether there exist further subclasses of ADFs, which can be calculated with a pure logic approach. Also it appears feasible to look for specific time semantics, e.g. where the truth-value of an argument has the least changes over a given time period.

Acknowledgement

We thank DFG (Deutsche Forschungsgemeinschaft) and BMBF (Bundesministerium für Bildung und Forschung) for funding this work (project numbers 406289255 and 01IS18026B).

References

[1] Bench-Capon TJM, Dunne PE. Argumentation in Artificial Intelligence. Artificial Intelligence. 2007;171(10-15):619–641.
[2] Baroni P, Gabbay D, Giacomin M, van der Torre L. Handbook of Formal Argumentation. College Publications; 2018.
[3] Dung PM. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artificial Intelligence. 1995;77(2):321–358.
[4] Baroni P, Caminada M, Giacomin M. Abstract Argumentation Frameworks and Their Semantics. In: Handbook of Formal Argumentation. College Publications; 2018. .
[5] Toni F. A tutorial on assumption-based argumentation. Argument & Computation. 2014;5(1):89–117.
[6] Modgil S, Prakken H. The $ASPIC^{+}$ framework for structured argumentation: a tutorial. Argument & Computation. 2014;5(1):31–62.
[7] Atkinson K, Baroni P, Giacomin M, Hunter A, Prakken H, Reed C, et al. Towards Artificial Argumentation. AI Magazine. 2017;38(3):25–36.
[8] Amgoud L, Vesic S. A new approach for preference-based argumentation frameworks. Ann Math Artif Intell. 2011;63(2):149–183.
[9] Bench-Capon TJM, Atkinson K. Abstract Argumentation and Values. In: Argumentation in Artificial Intelligence; 2009. p. 45–64.
[10] Nielsen SH, Parsons S. A Generalization of Dung's Abstract Framework for Argumentation: Arguing with Sets of Attacking Arguments. In: Workshop on Argumentation in Multi-Agent Systems; 2006. p. 54–73.
[11] Baroni P, Cerutti F, Giacomin M, Guida G. Encompassing Attacks to Attacks in Abstract Argumentation Frameworks. In: European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty; 2009. p. 83–94.
[12] Cayrol C, Lagasquie-Schiex M. Bipolar abstract argumentation systems. In: Argumentation in Artificial Intelligence; 2009. p. 65–84.
[13] Brewka G, Ellmauthaler S, Strass H, Wallner JP, Woltran S. Abstract Dialectical Frameworks Revisited. In: Rossi F, editor. Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013). IJCAI/AAAI; 2013. .
[14] Brewka G, Woltran S. Abstract Dialectical Frameworks. In: Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010; 2010. .
[15] Brewka G, Ellmauthaler S, Strass H, Wallner JP, Woltran S. Abstract Dialectical Frameworks. An Overview. IfCoLog Journal of Logics and their Applications Volume 4, number 8 Formal Argumentation. 2017 10;4(8):2263–2317.
[16] Denecker M, Marek VW, Truszczyński M. Ultimate approximation and its application in nonmonotonic knowledge representation systems. Inf Comput. 2004;192(1):84–121.
[17] Wintein S. On All Strong Kleene Generalizations of Classical Logic. Studia Logica. 2016;104(3):503–545.
[18] Budán MCD, Cobo ML, Martinez DC, Simari GR. Bipolarity in temporal argumentation frameworks. International Journal of Approximate Reasoning. 2017;84:1–22.
[19] Beck H, Dao-Tran M, Eiter T, Fink M. LARS: A Logic-Based Framework for Analyzing Reasoning over Streams. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. AAAI'15. AAAI Press; 2015. p. 1431–1438.