Argumentation among Agents: Review and Commentary

Grigore Costin-Teodor Radu Ștefan-Octavian
Vasiliu Florin Vintilă Eduard

Introduction

- Reference: Iyad Rahwan's *Argumentation among Agents*, Chapter 5 in *Multiagent Systems*, by G. Weiss.

Introduction

- Reference: Iyad Rahwan's *Argumentation among Agents*, Chapter 5 in *Multiagent Systems*, by G. Weiss.
- Our contribution: several new examples, and proofs for some merely stated claims.

Introduction

- Reference: Iyad Rahwan's *Argumentation among Agents*, Chapter 5 in *Multiagent Systems*, by G. Weiss.
- Our contribution: several new examples, and proofs for some merely stated claims.
- What is the author attempting to formalize?

Introduction

- Reference: Iyad Rahwan's *Argumentation among Agents*, Chapter 5 in *Multiagent Systems*, by G. Weiss.
- Our contribution: several new examples, and proofs for some merely stated claims.
- What is the author attempting to formalize?
- The philosopher's view of argumentation: the giving of claims in favor or against a statement that is open for debate.

Prakken's framework, briefly

- Idea: generalize common logics by admitting two kinds of inference rules – *strict* and *defeasible*.

Prakken's framework, briefly

- Idea: generalize common logics by admitting two kinds of inference rules – *strict* and *defeasible*.
- An *argumentation system* is a tuple $(\mathcal{L}, cont, S, D)$.

Prakken's framework, briefly

- Idea: generalize common logics by admitting two kinds of inference rules – *strict* and *defeasible*.
- An *argumentation system* is a tuple $(\mathcal{L}, cont, S, D)$.
- $\mathcal{L}$ is some "logical language" (must contain $\neg$).

Prakken's framework, briefly

- Idea: generalize common logics by admitting two kinds of inference rules – *strict* and *defeasible*.
- An *argumentation system* is a tuple $(\mathcal{L}, cont, S, D)$.
- $\mathcal{L}$ is some "logical language" (must contain $\neg$).
- The function

$$cont : \mathcal{L} \to \mathcal{P}(\mathcal{L})$$

  generalizes negation.

Prakken's framework, briefly

- Idea: generalize common logics by admitting two kinds of inference rules – *strict* and *defeasible*.
- An *argumentation system* is a tuple $(\mathcal{L}, cont, S, D)$.
- $\mathcal{L}$ is some "logical language" (must contain $\neg$).
- The function
$$cont : \mathcal{L} \to \mathcal{P}(\mathcal{L})$$
generalizes negation.
- S, D are respectively the sets of strict/defeasible inference rules.

Prakken's framework, briefly

- How does the *cont* function generalize negation?

Prakken's framework, briefly

- How does the *cont* function generalize negation?
- If $\varphi \in cont(\psi)$, then
  - if $\psi \notin cont(\varphi)$, then φ is a *contrary* of ψ;
  - if $\psi \in cont(\varphi)$, then φ and ψ are *contradictory*.

Prakken's framework, briefly

- How does the *cont* function generalize negation?
- If $\varphi \in cont(\psi)$, then
  - if $\psi \notin cont(\varphi)$, then φ is a *contrary* of ψ;
  - if $\psi \in cont(\varphi)$, then φ and ψ are *contradictory*.
- It is mandatory that

$$\neg\varphi \in cont(\varphi) \quad \text{and} \quad \varphi \in cont(\neg\varphi)$$

for any formula φ.

Prakken's framework, briefly

- An *argument* from a knowledge base $\mathcal{K}$ is defined similarly to a deduction in propositional logic. (The members of $\mathcal{K}$ play the role of the hypotheses.)

Prakken's framework, briefly

- An *argument* from a knowledge base $\mathcal{K}$ is defined similarly to a deduction in propositional logic. (The members of $\mathcal{K}$ play the role of the hypotheses.)
- Major difference: incorporation of the used inference rules.

Prakken's framework, briefly

- An *argument* from a knowledge base $\mathcal{K}$ is defined similarly to a deduction in propositional logic. (The members of $\mathcal{K}$ play the role of the hypotheses.)
- Major difference: incorporation of the used inference rules.
- The complete framework contains a partial order on defeasible rules. Using it, arguments may be compared.

Dung's model

- Henceforth, an *argumentation framework* will mean a finite directed graph $(\mathcal{A}, \rightharpoonup)$, whose nodes are called "arguments". The adjacency relation is pronounced "defeats".

Dung's model

- Henceforth, an *argumentation framework* will mean a finite directed graph $(\mathcal{A}, \rightharpoonup)$, whose nodes are called "arguments". The adjacency relation is pronounced "defeats".
- Hence, for arguments p, q, "$p \rightharpoonup q$" means "p defeats q".

Dung's model

- Henceforth, an *argumentation framework* will mean a finite directed graph $(\mathcal{A}, \rightharpoonup)$, whose nodes are called "arguments". The adjacency relation is pronounced "defeats".
- Hence, for arguments p, q, "$p \rightharpoonup q$" means "p defeats q".
- Note how the structure of arguments is not taken into account anymore.

Dung's model

- Henceforth, an *argumentation framework* will mean a finite directed graph $(\mathcal{A}, \rightharpoonup)$, whose nodes are called "arguments". The adjacency relation is pronounced "defeats".
- Hence, for arguments p, q, "$p \rightharpoonup q$" means "p defeats q".
- Note how the structure of arguments is not taken into account anymore.
- Objective: define an "acceptable" argument.

Dung's model

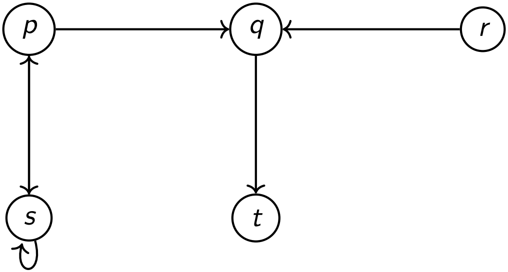

Figure: Our argumentation framework.

- $S^+ =$ the set of arguments defeated by some member of S.

Dung's model

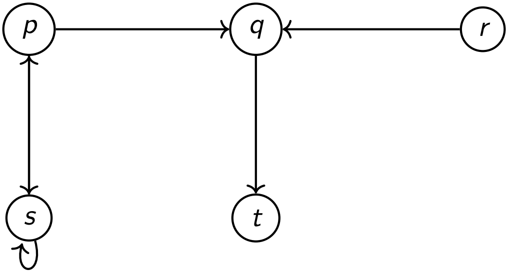

Figure: Our argumentation framework.

- $S^+ =$ the set of arguments defeated by some member of S.
- In the figure, $\{p, q\}^+ = \{q, s, t\}$.

Dung's model

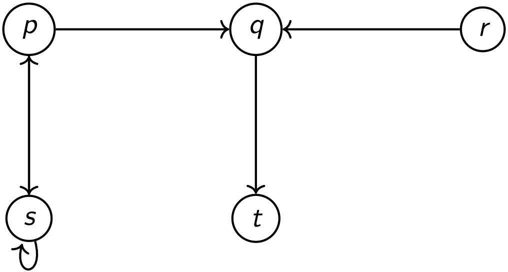

Figure: Our argumentation framework.

- $a^- =$ the set of arguments which defeat a.

Dung's model

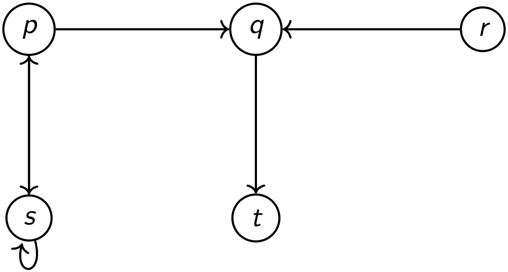

Figure: Our argumentation framework.

- a^{-} = the set of arguments which defeat a.
- In the figure, $s^{-} = \{p, s\}$.

Dung's model

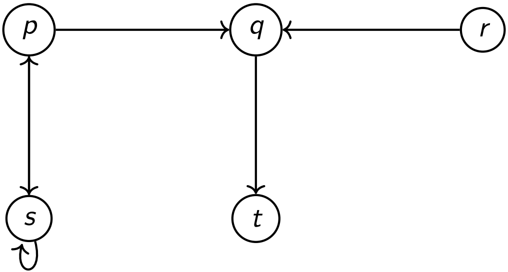

Figure: Our argumentation framework.

- A set S of arguments is *conflict-free* if no argument in S defeats another also in S.

Dung's model

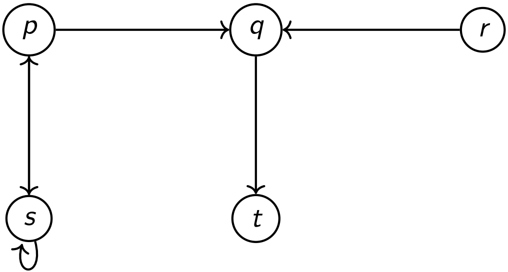

Figure: Our argumentation framework.

- A set S of arguments is *conflict-free* if no argument in S defeats another also in S.
- In the figure, $\{p, t\}$ and $\{r, t\}$ are conflict-free.

Dung's model

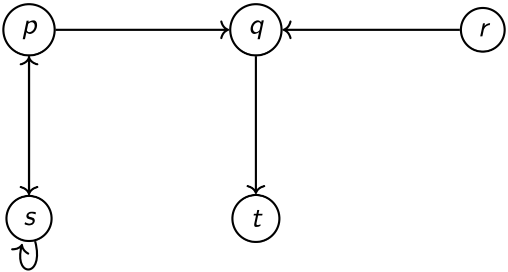

Figure: Our argumentation framework.

- A set S of arguments *defends* argument a if every argument which defeats a is defeated by S (i.e., is in S^+).

Dung's model

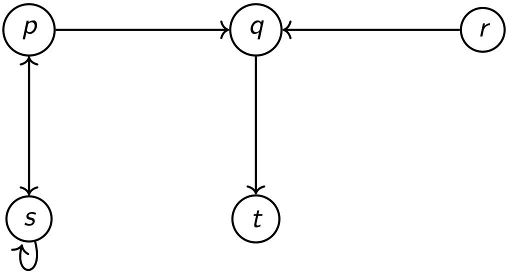

Figure: Our argumentation framework.

- A set S of arguments *defends* argument a if every argument which defeats a is defeated by S (i.e., is in S^+).
- In the figure, $\{p, t\}$ defends p.

Dung's model

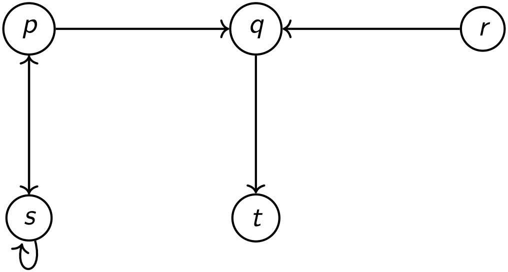

Figure: Our argumentation framework.

- The *characteristic function* $\mathcal{F}$ is defined thus:

$$\mathcal{F}(S) = \text{ the set of arguments defended by } S.$$

Dung's model

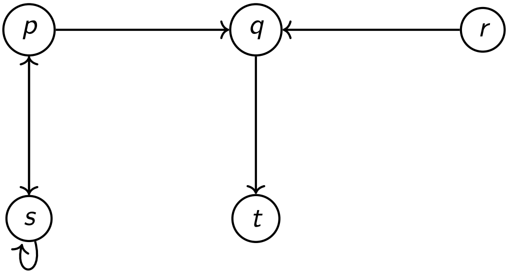

Figure: Our argumentation framework.

- The *characteristic function* $\mathcal{F}$ is defined thus:

$$\mathcal{F}(S) = \text{ the set of arguments defended by } S.$$

- In the figure, $\mathcal{F}(\{p, q, r\}) = \{p, r, t\}$ and $\mathcal{F}(\{r, t\}) = \{r, t\}$.

Dung's model

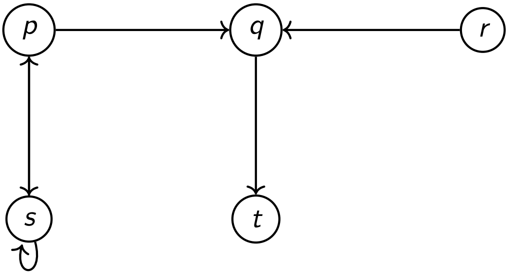

Figure: Our argumentation framework.

- A *complete extension* is a set S of arguments which is conflict-free and such that $\mathcal{F}(S) = S$ (i.e., it defends its own members and nothing else).

Dung's model

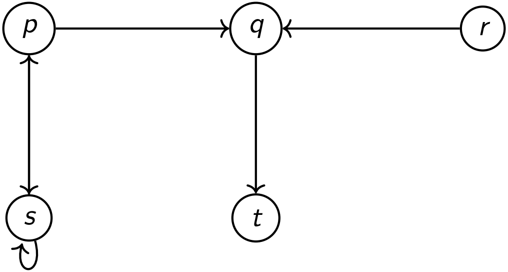

Figure: Our argumentation framework.

- A *complete extension* is a set S of arguments which is conflict-free and such that $\mathcal{F}(S) = S$ (i.e., it defends its own members and nothing else).
- By the remarks on previous slides, $\{r, t\}$ is a complete extension.

Dung's model

- The author exhibits an equivalent characterization of complete extensions via *labellings*.

Dung's model

- The author exhibits an equivalent characterization of complete extensions via *labellings*.
- An argument p is:
  - *skeptically accepted* iff p belongs to every extension;
  - *credulously accepted* iff p belongs to some extension;
  - *rejected* iff p doesn't belong to any extension.

Argumentation Games

- The author focuses on Dung's model to present a mechanism by which two agents can participate in a *dispute* where they can state and attack each other's arguments, much as in a real world debate.

Argumentation Games

- The author focuses on Dung's model to present a mechanism by which two agents can participate in a *dispute* where they can state and attack each other's arguments, much as in a real world debate.
- Formalize such an argumentation process and additionally enforce a set of constraints in order to capture various semantics (for example, an agent cannot contradict himself).

Argumentation Games

- Two players: **PRO** and **OPP**

Argumentation Games

- Two players: **PRO** and **OPP**
- PRO is the proponent which states the initial argument.

Argumentation Games

- Two players: **PRO** and **OPP**
- PRO is the proponent which states the initial argument.
- OPP is the opponent which begins by counter-attacking the argument proposed by PRO.

Argumentation Games

- Two players: **PRO** and **OPP**
- PRO is the proponent which states the initial argument.
- OPP is the opponent which begins by counter-attacking the argument proposed by PRO.
- Both players take turns in defeating the last argument that has been put forward by their counterpart player.

Argumentation Games

- Two players: **PRO** and **OPP**
- PRO is the proponent which states the initial argument.
- OPP is the opponent which begins by counter-attacking the argument proposed by PRO.
- Both players take turns in defeating the last argument that has been put forward by their counterpart player.
- The game is considered to be won by the player who states an argument a that cannot be defeated (i.e. $a^- = \emptyset$)

What is a dispute?

- The author calls the sequence of moves done by the players a *dispute*, a notion which is intensively used in further definitions and proofs.

What is a dispute?

- The author calls the sequence of moves done by the players a *dispute*, a notion which is intensively used in further definitions and proofs.
- However, a concrete definition is not provided. We attempt to state the following formal definition:

What is a dispute?

- The author calls the sequence of moves done by the players a *dispute*, a notion which is intensively used in further definitions and proofs.
- However, a concrete definition is not provided. We attempt to state the following formal definition:

Definition (dispute)

Given an argumentation framework $(\mathcal{A}, \rightharpoonup)$, a dispute is a sequence $(a_k)_{k \in \mathbb{N}}$ (possibly infinite) of arguments from $\mathcal{A}$ where the following property holds: $\forall i \in \mathbb{N}^*, a_i \rightharpoonup a_{i-1}$ (i.e. every argument in the sequence defeats its preceding argument).

Dispute trees

- Observation: A player could potentially counter-attack its counterpart player with any argument whatsoever that defeats the last move.

Dispute trees

- Observation: A player could potentially counter-attack its counterpart player with any argument whatsoever that defeats the last move.
- This leads to multiple disputes based on the defeating argument chosen by the player, which can be conveniently modeled as a *dispute tree*, as shown by the author.

Dispute trees

- Observation: A player could potentially counter-attack its counterpart player with any argument whatsoever that defeats the last move.
- This leads to multiple disputes based on the defeating argument chosen by the player, which can be conveniently modeled as a *dispute tree*, as shown by the author.
- Again, a definition is not provided. We attempt to adapt one from *Modgil et. al*:

Dispute trees

- Observation: A player could potentially counter-attack its counterpart player with any argument whatsoever that defeats the last move.
- This leads to multiple disputes based on the defeating argument chosen by the player, which can be conveniently modeled as a *dispute tree*, as shown by the author.
- Again, a definition is not provided. We attempt to adapt one from *Modgil et. al*:

Definition (dispute trees)

Given an argumentation framework $(\mathcal{A}, \rightharpoonup)$ and an argument p in $\mathcal{A}$, a dispute tree induced by p is a tree T rooted in p, where each node represents an argument from $\mathcal{A}$ and for all arguments x, y in $\mathcal{A}$, x is a child of y in T iff x defeats y.

Protocol G

- The author establishes a rule (called protocol G) by which the PRO player cannot repeat an argument in a dispute.

Protocol G

- The author establishes a rule (called protocol G) by which the PRO player cannot repeat an argument in a dispute.
- We provide a definition with regards to a decision tree:

Protocol G

- The author establishes a rule (called protocol G) by which the PRO player cannot repeat an argument in a dispute.
- We provide a definition with regards to a decision tree:

Definition (dispute tree under protocol G)

Given a dispute tree T, we consider T to be under protocol G iff for any arbitrary dispute d in T and for any pair of arguments x, y stated by PRO in d, x is different than y.

Protocol G

- The author establishes a rule (called protocol G) by which the PRO player cannot repeat an argument in a dispute.
- We provide a definition with regards to a decision tree:

Definition (dispute tree under protocol G)

Given a dispute tree T, we consider T to be under protocol G iff for any arbitrary dispute d in T and for any pair of arguments x, y stated by PRO in d, x is different than y.

- It is claimed by the author that the following property is true, to which we provide a proof:

Protocol G

- The author establishes a rule (called protocol G) by which the PRO player cannot repeat an argument in a dispute.
- We provide a definition with regards to a decision tree:

Definition (dispute tree under protocol G)

Given a dispute tree T, we consider T to be under protocol G iff for any arbitrary dispute d in T and for any pair of arguments x, y stated by PRO in d, x is different than y.

- It is claimed by the author that the following property is true, to which we provide a proof:

Claim

If T is a dispute tree under protocol G, then T is finite.

Proof sketch

- Let $n = card(\mathcal{A})$ and $d = (a_k)_{k \in \mathbb{N}}$ be a dispute in T of length of at least $2n$ arguments (we do not consider the other disputes, since we know they are of finite length).

Proof sketch

- Let $n = card(\mathcal{A})$ and $d = (a_k)_{k\in\mathbb{N}}$ be a dispute in T of length of at least $2n$ arguments (we do not consider the other disputes, since we know they are of finite length).
- We shall prove that d is of finite length; more specifically, exactly of length $2n$. We consider the argument a_{2n-1} from our sequence d.

Proof sketch

- Let $n = card(\mathcal{A})$ and $d = (a_k)_{k \in \mathbb{N}}$ be a dispute in T of length of at least $2n$ arguments (we do not consider the other disputes, since we know they are of finite length).
- We shall prove that d is of finite length; more specifically, exactly of length $2n$. We consider the argument a_{2n-1} from our sequence d.
- By protocol G, we have exactly n different arguments uttered by PRO, which cover all the arguments in the set $\mathcal{A}$.

Proof sketch

- Let $n = card(\mathcal{A})$ and $d = (a_k)_{k\in\mathbb{N}}$ be a dispute in T of length of at least $2n$ arguments (we do not consider the other disputes, since we know they are of finite length).
- We shall prove that d is of finite length; more specifically, exactly of length $2n$. We consider the argument a_{2n-1} from our sequence d.
- By protocol G, we have exactly n different arguments uttered by PRO, which cover all the arguments in the set $\mathcal{A}$.
- Assume that a_{2n} exists. This being the $n+1$'th argument stated by PRO, it must coincide with an argument that has been uttered before, since we have a total of only n different arguments to choose from (Dirichlet's box principle).

Proof sketch

- Let $n = card(\mathcal{A})$ and $d = (a_k)_{k \in \mathbb{N}}$ be a dispute in T of length of at least $2n$ arguments (we do not consider the other disputes, since we know they are of finite length).
- We shall prove that d is of finite length; more specifically, exactly of length $2n$. We consider the argument a_{2n-1} from our sequence d.
- By protocol G, we have exactly n different arguments uttered by PRO, which cover all the arguments in the set $\mathcal{A}$.
- Assume that a_{2n} exists. This being the $n+1$'th argument stated by PRO, it must coincide with an argument that has been uttered before, since we have a total of only n different arguments to choose from (Dirichlet's box principle).
- However, this contradicts protocol G. Hence, d is finite.