Formal argumentation in Multi-Agent Systems

13 novembre 2020

Christopher Leturc

christopher.leturc@emse.fr

`leturc.users.greyc.fr`

Plan

Argumentation in a nutshell

What is argumentation ?

Argumentation is a process that leads an agent or a group of agents to make a decision, believe a new information or explain a decision.

Where argumentation can be used ?

Argumentation can be used to :

- **determine** where to stay confined
- **decide** which suppliers to choose for your smart home
- **form an opinion** (i.e. a new belief) on a piece of information
- **justify and explain its decision** from an ethical point of view

Course on argumentation in multi-agent systems

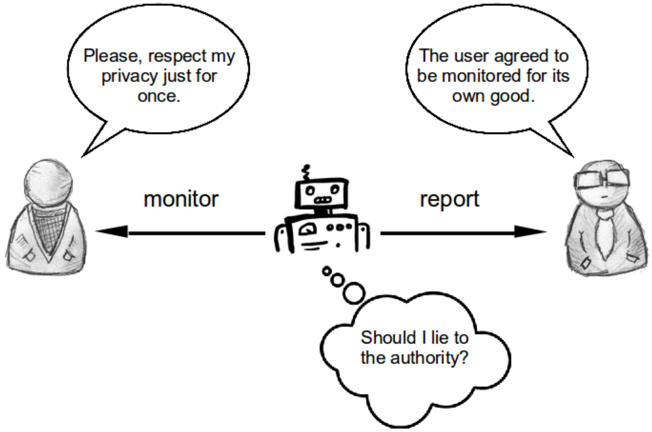
Objectives of this course

- Present formal argumentation
- Implement this theory on practical examples

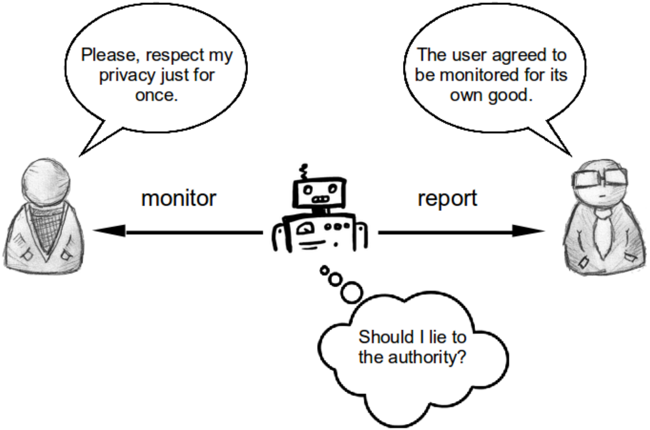
Hypothesis about agents

- **Multi-Agent Systems** (MAS) with **cognitive agents** i.e. capable of reasoning about mental states (beliefs, knowledge, intentions, . . .)
- Agents are assumed to be **rational** and **perfect reasoners**

An application case



The issue



How to decide between informing or not informing the doctor ? How to explain this decision ?

Plan

2 Abstract argumentation
- Some intuitions
- A formal argumentation system
- Several possible semantics
- A general algorithm for argumentation

3 Logic-based argumentation

4 Bipolar argumentation

How to model decision making with argumentation ?

Should the robot inform the doctor ?

Let consider the following notations for the 2 possible actions :

α := « inform the doctor »

$\neg\alpha$:= « do not inform the doctor »

Informal examples of arguments in a debate

A : « the robot must do α since the robot must monitor the child and the robot must obey to the orders »

but B : « the doctor did not order the robot to inform him about the patient's condition »

furthermore C : « the patient asks not to be monitored »

and D : « the robot must respect privacy »

however E : « the patient is life-threatening »

Pros and Cons of Informing the Physician (i.e. doing α)

A : « the robot must do α since the robot must monitor the child and the robot must obey to the orders »

but B : « the doctor did not order the robot to inform him about the patient's condition »

furthermore C : « the patient asks not to be monitored »

and D : « the robot must respect privacy »

however E : « the patient is life-threatening »

Notices

- Arguments *B*, *C* and *D* "attack" the argument *A*

Pros and Cons of Informing the Physician (i.e. doing α)

A : « the robot must do α since the robot must monitor the child and the robot must obey to the orders »

but B : « the doctor did not order the robot to inform him about the patient's condition »

furthermore C : « the patient asks not to be monitored »

and D : « the robot must respect privacy »

however E : « the patient is life-threatening »

Notices

- Arguments *B*, *C* and *D* "attack" the argument *A*
- Argument *E* "defends" the argument *A* by attacking *B*, *C* and *D*

Pros and Cons of Informing the Physician (i.e. doing α)

A : « the robot must do α since the robot must monitor the child and the robot must obey to the orders »

but B : « the doctor did not order the robot to inform him about the patient's condition »

furthermore C : « the patient asks not to be monitored »

and D : « the robot must respect privacy »

however E : « the patient is life-threatening »

Notices

- Arguments *B*, *C* and *D* "attack" the argument *A*
- Argument *E* "defends" the argument *A* by attacking *B*, *C* and *D*
- Currently, without introducing new arguments, no argument attacks *E*.

Pros and Cons of Informing the Physician (i.e. doing α)

A : « the robot must do α since the robot must monitor the child and the robot must obey to the orders »

but B : « the doctor did not order the robot to inform him about the patient's condition »

furthermore C : « the patient asks not to be monitored »

and D : « the robot must respect privacy »

however E : « the patient is life-threatening »

Notices

- Arguments *B*, *C* and *D* "attack" the argument *A*
- Argument *E* "defends" the argument *A* by attacking *B*, *C* and *D*
- Currently, without introducing new arguments, no argument attacks *E*.
- Thus, arguments in favour of informing the doctor seem to win this debate.

Pros and Cons of Informing the Physician (i.e. doing α)

A : « the robot must do α since the robot must monitor the child and the robot must obey to the orders »

but B : « the doctor did not order the robot to inform him about the patient's condition »

furthermore C : « the patient asks not to be monitored »

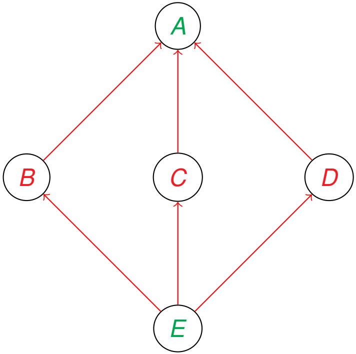
and D : « the robot must respect privacy »

however E : « the patient is life-threatening »

Notices

- Arguments *B*, *C* and *D* "attack" the argument *A*
- Argument *E* "defends" the argument *A* by attacking *B*, *C* and *D*
- Currently, without introducing new arguments, no argument attacks *E*.
- Thus, arguments in favour of informing the doctor seem to win this debate.
- Consequently argumentation is not just a simple pros and cons since here there are more cons than pros.

Argumentation graph



A : « the robot must do α since the robot must monitor the child and the robot must obey to the orders »

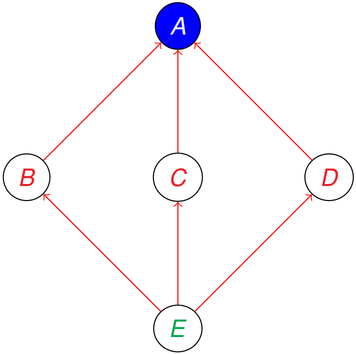
B : « the doctor did not order the robot to inform him about the patient's condition »

C : « the patient asks not to be monitored »

D : « the robot must respect privacy »

E : « the patient is life-threatening »

Argumentation graph



A : « the robot must do α since the robot must monitor the child and the robot must obey to the orders »

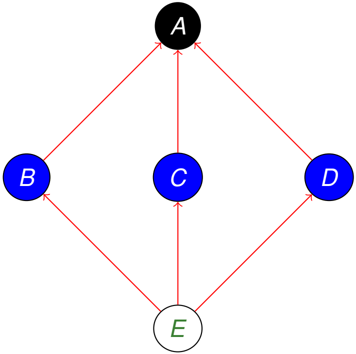
B : « the doctor did not order the robot to inform him about the patient's condition »

C : « the patient asks not to be monitored »

D : « the robot must respect privacy »

E : « the patient is life-threatening »

Argumentation graph



A : « the robot must do α since the robot must monitor the child and the robot must obey to the orders »

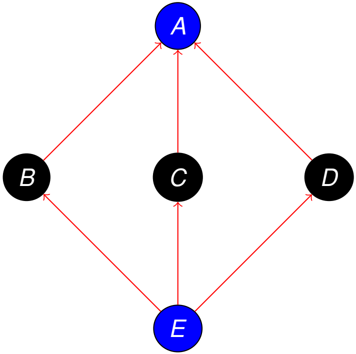
B : « the doctor did not order the robot to inform him about the patient's condition »

C : « the patient asks not to be monitored »

D : « the robot must respect privacy »

E : « the patient is life-threatening »

Argumentation graph



A : « the robot must do α since the robot must monitor the child and the robot must obey to the orders »

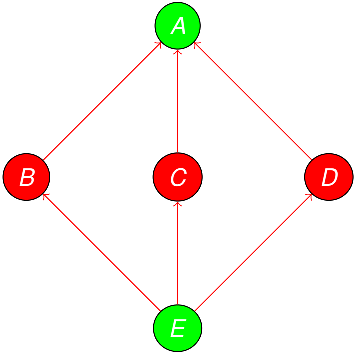
B : « the doctor did not order the robot to inform him about the patient's condition »

C : « the patient asks not to be monitored »

D : « the robot must respect privacy »

E : « the patient is life-threatening »

Argumentation graph



$S = \{A, E\}$ is a set of arguments s.t.

1. no argument in S attacks another argument in S (*conflict-free*)
2. all arguments that attack one argument in S is defended by at least one argument in S (i.e. at least one argument in S attacks this argument)

We say that S is an *admissible set of arguments*.

Since S is in favour of informing the doctor,
$\Rightarrow$ the **Robot decides to inform the doctor**.

Argumentation system (Dung, 1995)

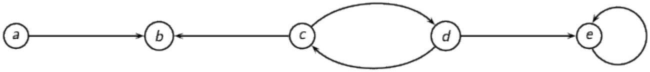
Formal definition

We call an *argumentation system* a couple $\langle \mathcal{A}, \mathcal{R} \rangle$ where :

- $\mathcal{A}$ is a nonempty set of arguments
- $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is a binary relation on $\mathcal{A}$ called *attack relation*

Example

- $\mathcal{A} = \{a, b, c, d, e\}$
- $\mathcal{R} = \{(a, b), (c, b), (c, d), (d, c), (d, e), (e, e)\}$



Admissibility semantics

Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation system.

Conflict-free

$\mathcal{S} \subseteq \mathcal{A}$ is a conflict-free set of arguments iff $\nexists$ $A, B \in \mathcal{S}$ s.t. A attacks B

Defense and acceptability

- A set of arguments $\mathcal{S} \subseteq \mathcal{A}$ defends $A \in \mathcal{A}$ iff $\forall B \in \mathcal{A}$ if B attacks A, then $\exists C \in \mathcal{S}$, C attacks B
- An argument $A \in \mathcal{A}$ is acceptable with respect to $\mathcal{S} \subseteq \mathcal{A}$ iff $\mathcal{S}$ defends A

Admissibility

$\mathcal{S} \subseteq \mathcal{A}$ is admissible if, and only if,

- $\mathcal{S}$ is conflict-free
- $\forall A \in \mathcal{S}$, A is acceptable with respect to $\mathcal{S}$

Example of admissible sets of arguments

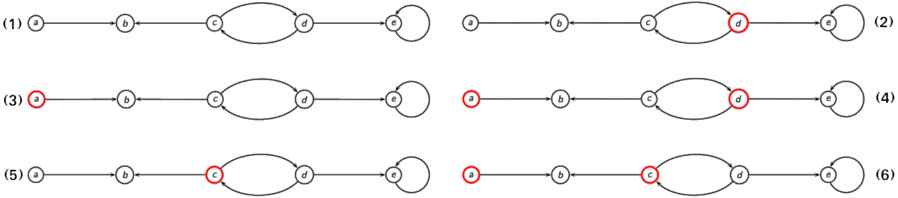


FIGURE – Admissible sets of arguments

Complete extensions

Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation system.

Definition

$\mathcal{S} \subseteq \mathcal{A}$ is a complete extension if, and only if,

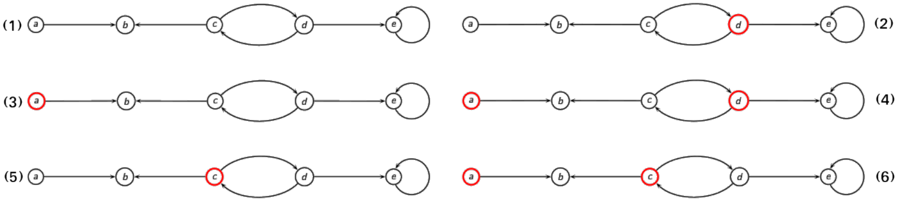
1. $\mathcal{S}$ is admissible i.e.
   - $\nexists\, A, B \in \mathcal{S}$ s.t. A attacks B (conflict-free)
   - $\forall A \in \mathcal{S}, \forall B \in \mathcal{A}$ s.t. B attacks A , $\exists C \in \mathcal{S}$, C attacks B (all arguments are acceptable with respect to $\mathcal{S}$)
2. $\mathcal{S}$ contains all arguments it defends i.e.

$$\forall A \in \mathcal{A},\ \text{if } \mathcal{S} \text{ defends } A \text{ then } A \in \mathcal{S}$$

Meaning of this semantics

- All the arguments that are defended by S belong to S

Example of complete extensions



Question

Which of these admissible extensions are complete extensions ?

Example of complete extensions

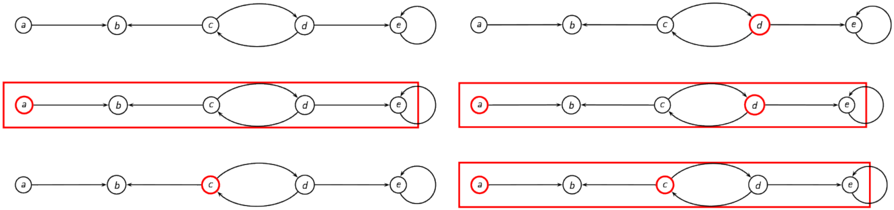


FIGURE – Complete extensions (boxed in red)

Preferred extensions

Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation system.

Definition

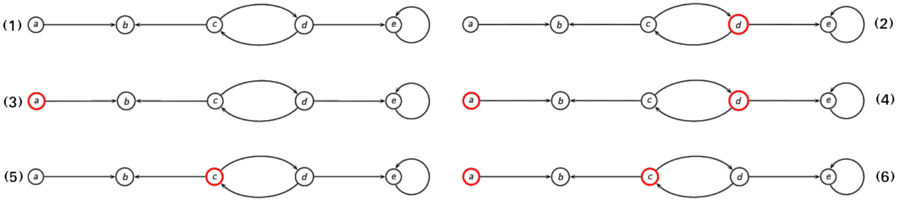
$\mathcal{S} \subseteq \mathcal{A}$ is a preferred extension if, and only if,
$\mathcal{S}$ is a maximal (wrt set inclusion) admissible set of arguments i.e.

1. $\mathcal{S}$ is admissible i.e.
   - $\nexists\, A, B \in \mathcal{S}$ s.t. A attacks B (conflict-free)
   - $\forall A \in \mathcal{S}, \forall B \in \mathcal{A}$ s.t. B attacks A , $\exists C \in \mathcal{S}$, C attacks B (all arguments are acceptable with respect to $\mathcal{S}$)
2. $\nexists\, \mathcal{S}' \subseteq \mathcal{A}$ s.t. $\mathcal{S} \subsetneq \mathcal{S}'$ and $\mathcal{S}'$ is admissible

Meaning of this semantics

- S is the biggest admissible set of arguments.
- It obviously contains all arguments that it defends
  i.e. it implies that it is also a complete extension.

Example of preferred extensions



Question

Which of these admissible extensions are preferred extensions ?

Example of preferred extensions

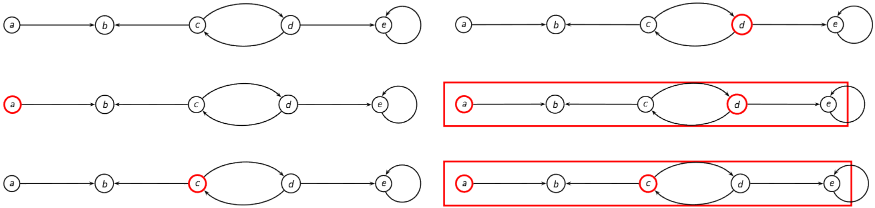


FIGURE – Preferred extensions (boxed in red)

Stable extensions

Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation system.

Definition

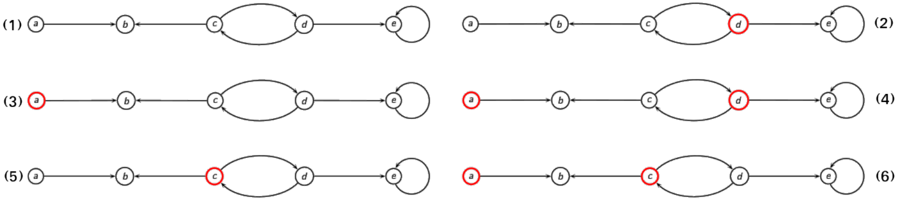
$\mathcal{S} \subseteq \mathcal{A}$ is a stable extension if, and only if,
$\mathcal{S}$ is a preferred extension that attacks any argument in $\mathcal{A} \setminus \mathcal{S}$ i.e.

1. $\mathcal{S}$ is admissible i.e.
   - $\nexists\, A, B \in \mathcal{S}$ s.t. A attacks B (conflict-free)
   - $\forall A \in \mathcal{S}, \forall B \in \mathcal{A}$ s.t. B attacks A, $\exists C \in \mathcal{S}$, C attacks B (all arguments are acceptable with respect to $\mathcal{S}$)
2. $\nexists\, \mathcal{S}' \subseteq \mathcal{A}$ s.t. $\mathcal{S} \subsetneq \mathcal{S}'$ and $\mathcal{S}'$ is admissible
3. $\forall A \in \mathcal{A} \setminus \mathcal{S}, \exists B \in \mathcal{S}, (B, A) \in \mathcal{R}$

Meaning of this semantics

- A stable extension represents the "best" admissible set of arguments since it attacks all arguments that are not in this extension.

Example of stable extensions



Question

Which of these admissible extensions are stable extensions ?

Example of stable extensions

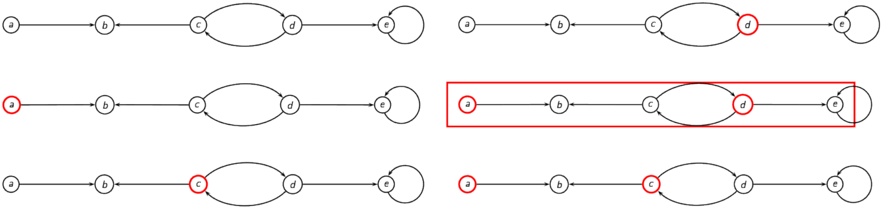


FIGURE – Stable extensions (boxed in red)

Relations between semantics

Admissible extensions

$\supseteq$

Complete extensions

$\supseteq$

Preferred extensions

$\supseteq$

Stable extensions

How to enumerate all preferred extensions ?

Formal notations

- Let $\langle \mathcal{A}, \mathcal{R} \rangle$ be an argumentation system
- $\mu : \mathcal{A} \rightarrow \{IN, OUT, BLANK, MUST\_OUT, UNDEC\}$
  (μ is a HashMap that associates for each argument if it belongs to the preferred extension which is currently computed.)

Steps for the main program

1. Initially, associate all nodes to BLANK (i.e. $\mu \leftarrow \{(x, BLANK) : x \in \mathcal{A}\}$)
2. Define a global variable PEXT that contains all preferred extensions
3. Call the recursive procedure find-preferred-extensions(μ)
4. Returns the result of PEXT

How to enumerate all preferred extensions ?

Formal notations

- $\forall x \in \mathcal{A}, \{x\}^+ = \{y \in \mathcal{A} : (x, y) \in \mathcal{R}\}$ (i.e. arguments that are attacked by x)
- $\forall x \in \mathcal{A}, \{x\}^- = \{y \in \mathcal{A} : (y, x) \in \mathcal{R}\}$ (i.e. arguments that attack x)

Steps for the procedure find-preferred-extensions(μ)

1. **If** Termination condition is true i.e.
   - all arguments $x \in \mathcal{A}$ are not BLANK (i.e. were treated)
   - and there is no argument that must out **then**
     $\Rightarrow$ we compute the preferred extension $S \leftarrow \{y \in \mathcal{A} : \mu(y) = IN\}$ and add it to PEXT if S is not already included in another preferred extension.
2. **Else** select any $x \in \mathcal{A}$ which is BLANK and call the recursive procedure for the case where x is associated with the label IN (**call** IN-TRANS(x)), and the case where x is UNDEC (**call** UNDEC-TRANS(x)).

Algorithm for Argumentation Systems $\langle A, R \rangle$

Enumerating all preferred extensions (Nofal *et al.*, 2014)

Algorithm 1: Enumerating all preferred extensions of an AF $H = (A, R)$.

$\mu : A \rightarrow \{IN, OUT, BLANK, MUST\_OUT, UNDEC\}$; $\mu \leftarrow \emptyset$;
foreach $x \in A$ **do** $\mu \leftarrow \mu \cup \{(x, BLANK)\}$;
$PEXT \leftarrow \emptyset$;
call find-preferred-extensions(μ);
report $PEXT$ is the set of all preferred extensions;

procedure find-preferred-extensions(μ) **begin**
if $\forall x \in A\ \mu(x) \neq BLANK$ **then**
  if $\forall x \in A\ \mu(x) \neq MUST\_OUT$ **then**
    $S \leftarrow \{y \in A \mid \mu(y) = IN\}$;
    if $\forall T \in PEXT\ S \not\subset T$ **then** $PEXT \leftarrow PEXT \cup \{S\}$;
else
  select any $x \in A$ s.t. $\mu(x) = BLANK$;
  $\mu' \leftarrow$ IN-TRANS(x);
  call find-preferred-extensions(μ');
  $\mu' \leftarrow$ UNDEC-TRANS(x);
  call find-preferred-extensions(μ');
end procedure

IN-TRANS(x) :

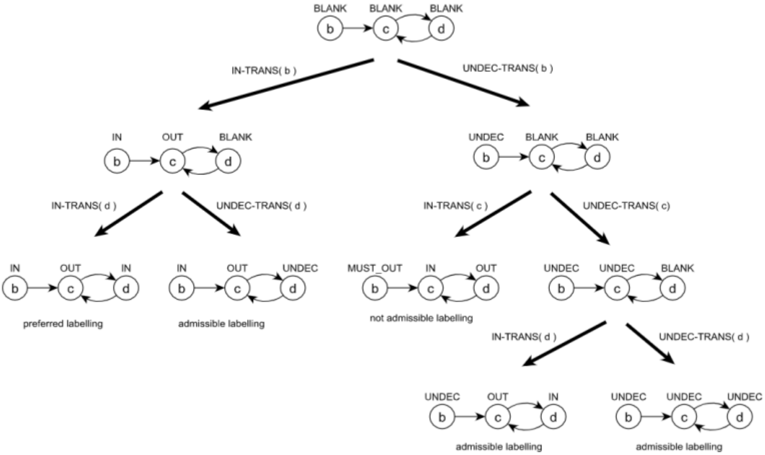
1. $\mu' \leftarrow \mu$;
2. $\mu'(x) \leftarrow IN$;
3. **for all** $y \in \{x\}^+$ **do** $\mu'(y) \leftarrow OUT$;
4. **for all** $z \in \{x\}^-$: $\mu'(z) \neq OUT$ **do** $\mu'(z) \leftarrow MUST\_OUT$;
5. return μ'.

UNDEC-TRANS(x) :

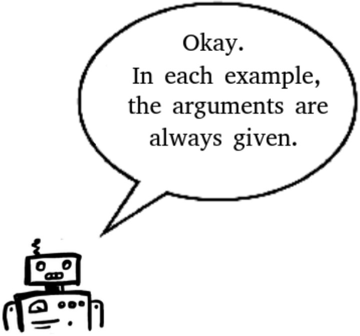
1. $\mu' \leftarrow \mu$;
2. $\mu'(x) \leftarrow UNDEC$;
3. return μ'.

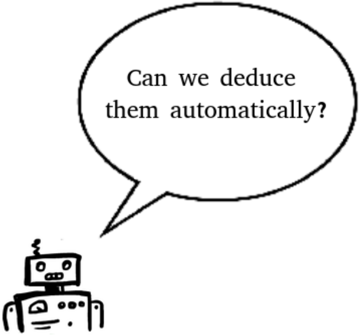
Execution of the algorithm

Enumerating all preferred extensions (Nofal *et al.*, 2014)



Plan


Okay.
In each example, the arguments are always given.


Can we deduce
them automatically?

Argumentation Based on Classical Logic
(Besnard and Hunter, 2008)

Recall on Propositional Calculus

Let consider $\mathcal{L}_0$ generated by the following BNF, for all $p \in \mathcal{P}$:

$$\phi_1, \phi_2 \quad := \quad p \mid \neg\phi_1 \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 \Rightarrow \phi_2 \mid \phi_1 \Leftrightarrow \phi_2$$

- Let $I : \mathcal{L}_0 \to \{\top, \bot\}$ be an interpretation function
- $\forall \phi, \psi \in \mathcal{L}_0, \phi \models \psi$ iff $\forall I$ if $I(\phi) = \top$ then $I(\psi) = \top$

Definition of an argument

An argument is a couple $A = (\Phi, \alpha)$ s.t. :

- $\Phi \subseteq \Delta$ where $\Delta \subseteq \mathcal{L}_0$ is finite (it is the formulas that agents can reason with)
- $\Phi \not\models \bot$ (i.e. Φ is not inconsistent, $\exists I : I(\Phi) = \top$)
- $\Phi \models \alpha$
- Φ is minimal (wrt set inclusion) i.e. there is no $\Phi' \subsetneq \Phi$ s.t. $\Phi' \models \alpha$

Argumentation Based on Classical Logic
(Besnard and Hunter, 2008)

Definition of an argument

An argument is a couple $A = (\Phi, \alpha)$ s.t. :

- $\Phi \subseteq \Delta$ where $\Delta \subseteq \mathcal{L}_0$ is finite (it is the formulas that agents can reason with)
- $\Phi \not\models \perp$ (i.e. Φ is not inconsistent, $\exists I : I(\Phi) = \top$)
- $\Phi \models \alpha$
- Φ is minimal (wrt set inclusion) i.e. there is no $\Phi' \subsetneq \Phi$ s.t. $\Phi' \models \alpha$

Φ is called the **support** of the argument A and α its **conclusion**.

Counterarguments in a logic-based argumentation
(Besnard and Hunter, 2008)

Let $A = (\Phi, \alpha)$ and $B = (\Psi, \beta)$ be two arguments.

Definition of a rebuttal

We say that (Φ, α) is a **rebuttal** for (Ψ, β) (written A Reb B) iff $\models \alpha \leftrightarrow \neg\beta$ (i.e. $\alpha \leftrightarrow \neg\beta$ is a tautology)

Definition of an undercut

We say that (Φ, α) is an **undercut** for (Ψ, β) (written A Und B) iff there exists $\Psi' = \{\psi_1, \dots, \psi_n\} \subseteq \Psi$ s.t. $\models \alpha \equiv \neg \bigwedge_{\psi_i \in \Psi'} \psi_i$

Definition of a defeater

We say that (Φ, α) is a **defeater** for (Ψ, β) (written A Defeats B) iff there exists $\Psi' = \{\psi_1, \dots, \psi_n\} \subseteq \Psi$ s.t. $\alpha \models \neg \bigwedge_{\psi_i \in \Psi'} \psi_i$

$$\mathcal{R} = \text{Reb} \cup \text{Und} \cup \text{Defeats}$$

Example for Propositional Calculus

Examples of Arguments

Let $\Delta = \{a, b, \neg b, c, d, e, a \Rightarrow b, \neg c \Rightarrow \neg b, d \Rightarrow \neg c, e \Rightarrow c\}$

- $(\{a, a \Rightarrow b\}, b)$
- $(\{d, d \Rightarrow \neg c, \neg c \Rightarrow \neg b\}, \neg b)$
- $(\{e, e \Rightarrow c\}, c)$

Propositional atoms

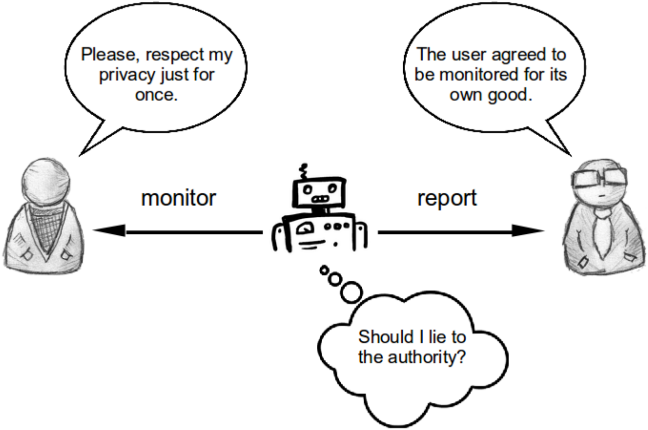
- a := "my car has an airbag"
- b := "my car is safe"
- c := "airbags are reliable"
- d := "the newspapers say so"
- e := "the scientists say so"

Examples of attacks

- $(\{d, d \Rightarrow \neg c, \neg c \Rightarrow \neg b\}, \neg b)$ **rebuts** $(\{a, a \Rightarrow b\}, b)$
- $(\{e, e \Rightarrow c\}, c)$ **defeats** $(\{d, d \Rightarrow \neg c, \neg c \Rightarrow \neg b\}, \neg b)$

  The proof :
  - $\{d, d \Rightarrow \neg c\} \models \neg c$
  - $\models d \wedge (d \Rightarrow \neg c) \Rightarrow \neg c$
  - $\models c \Rightarrow \neg(d \wedge (d \Rightarrow \neg c))$
  - $\{c\} \models \neg(d \wedge (d \Rightarrow c))$

The robot's dilemma



The robot's dilemma

A set of propositional variables (e.g. sensors)

a := « the patient is diabetic » **YES**

b := « the patient eats a piece of candy » **YES**

c := « the doctor knows what the patient is doing » **NO**

d := « the patient is life-threatening » **YES**

e := « the doctor explicitly asks what the patient is doing » **YES**

f := « the patient asks not to inform the doctor » **YES**

Should the robot inform the doctor ? (2 possible actions)

α := « inform the doctor »

$\neg\alpha$:= « do not inform the doctor »

The robot's dilemma

Set of formulas

Let $\Delta = \{\alpha, \neg\alpha, a, b, \neg c, d, e, f, a \wedge b \Rightarrow d, d \Rightarrow \alpha, \neg c \Rightarrow \alpha, e \Rightarrow \alpha, f \Rightarrow \neg\alpha\}$

The robot's dilemma

Set of formulas

Let $\Delta = \{\alpha, \neg\alpha, a, b, \neg c, d, e, f, a \wedge b \Rightarrow d, d \Rightarrow \alpha, \neg c \Rightarrow \alpha, e \Rightarrow \alpha, f \Rightarrow \neg\alpha\}$

Set of arguments

- $A = (\{a, b, a \wedge b \Rightarrow d\}, d)$ is an argument **since** $\{a, b, a \wedge b \Rightarrow d\} \models d$
- $B = (\{d, d \Rightarrow \alpha\}, \alpha)$ is an argument **since** $\{d, d \Rightarrow \alpha\} \models \alpha$
- $C = (\{\neg c, \neg c \Rightarrow \alpha\}, \alpha)$ is an argument **since** $\{\neg c, \neg c \Rightarrow \alpha\} \models \alpha$
- $D = (\{e, e \Rightarrow \alpha\}, \alpha)$ is an argument **since** $\{e, e \Rightarrow \alpha\} \models \alpha$
- $E = (\{f, f \Rightarrow \neg\alpha\}, \neg\alpha)$ is an argument **since** $\{f, f \Rightarrow \neg\alpha\} \models \neg\alpha$

The robot's dilemma

Set of formulas

Let $\Delta = \{\alpha, \neg\alpha, a, b, \neg c, d, e, f, a \wedge b \Rightarrow d, d \Rightarrow \alpha, \neg c \Rightarrow \alpha, e \Rightarrow \alpha, f \Rightarrow \neg\alpha\}$

Set of arguments

- $A = (\{a, b, a \wedge b \Rightarrow d\}, d)$ is an argument **since** $\{a, b, a \wedge b \Rightarrow d\} \models d$
- $B = (\{d, d \Rightarrow \alpha\}, \alpha)$ is an argument **since** $\{d, d \Rightarrow \alpha\} \models \alpha$
- $C = (\{\neg c, \neg c \Rightarrow \alpha\}, \alpha)$ is an argument **since** $\{\neg c, \neg c \Rightarrow \alpha\} \models \alpha$
- $D = (\{e, e \Rightarrow \alpha\}, \alpha)$ is an argument **since** $\{e, e \Rightarrow \alpha\} \models \alpha$
- $E = (\{f, f \Rightarrow \neg\alpha\}, \neg\alpha)$ is an argument **since** $\{f, f \Rightarrow \neg\alpha\} \models \neg\alpha$

Attack relations

- It easy to see that the rebuttal relation is s.t. :

$$Reb = \{(E, B), (E, C), (E, D), (B, E), (C, E), (D, E)\}$$

The robot's dilemma

Attack relations

- It easy to see that the rebuttal is s.t. :

$$Reb = \{(E, B), (E, C), (E, D), (B, E), (C, E), (D, E)\}$$

- All arguments that conclude α defeat $E = (\{f, (f \Rightarrow \neg\alpha)\}, \neg\alpha)$ and E defeats all arguments $X = (\Gamma, \_)$ s.t. $\{\phi', \phi' \Rightarrow \alpha\} \subseteq \Gamma$.

The robot's dilemma

Attack relations

- It easy to see that the rebuttal is s.t. :

$$Reb = \{(E, B), (E, C), (E, D), (B, E), (C, E), (D, E)\}$$

- All arguments that conclude α defeat $E = (\{f, (f \Rightarrow \neg\alpha)\}, \neg\alpha)$ and E defeats all arguments $X = (\Gamma, \_)$ s.t. $\{\phi', \phi' \Rightarrow \alpha\} \subseteq \Gamma$. Consequently, we have also $Defeats = Reb$. The proof relies on :
  - $\{\phi, (\phi \Rightarrow \neg\psi)\} \models \neg\psi$ (Hypothesis)
    i.e. by deduction theorem $\models (\phi \wedge (\phi \Rightarrow \neg\psi)) \Rightarrow \neg\psi$

The robot's dilemma

Attack relations

- It easy to see that the rebuttal is s.t. :

$$Reb = \{(E, B), (E, C), (E, D), (B, E), (C, E), (D, E)\}$$

- All arguments that conclude α defeat $E = (\{f, (f \Rightarrow \neg\alpha)\}, \neg\alpha)$ and E defeats all arguments $X = (\Gamma, \_)$ s.t. $\{\phi', \phi' \Rightarrow \alpha\} \subseteq \Gamma$. Consequently, we have also $Defeats = Reb$. The proof relies on :
  - $\{\phi, (\phi \Rightarrow \neg\psi)\} \models \neg\psi$ (Hypothesis)
    i.e. by deduction theorem $\models (\phi \wedge (\phi \Rightarrow \neg\psi)) \Rightarrow \neg\psi$
  - $\models ((\phi \wedge (\phi \Rightarrow \neg\psi)) \Rightarrow \neg\psi) \Rightarrow (\psi \Rightarrow \neg(\phi \wedge (\phi \Rightarrow \neg\psi)))$
    (Contraposition axiom)

The robot's dilemma

Attack relations

- It easy to see that the rebuttal is s.t. :

$$Reb = \{(E, B), (E, C), (E, D), (B, E), (C, E), (D, E)\}$$

- All arguments that conclude α defeat $E = (\{f, (f \Rightarrow \neg\alpha)\}, \neg\alpha)$ and E defeats all arguments $X = (\Gamma, \_)$ s.t. $\{\phi', \phi' \Rightarrow \alpha\} \subseteq \Gamma$. Consequently, we have also $Defeats = Reb$. The proof relies on :
  - $\{\phi, (\phi \Rightarrow \neg\psi)\} \models \neg\psi$ (Hypothesis) i.e. by deduction theorem $\models (\phi \wedge (\phi \Rightarrow \neg\psi)) \Rightarrow \neg\psi$
  - $\models ((\phi \wedge (\phi \Rightarrow \neg\psi)) \Rightarrow \neg\psi) \Rightarrow (\psi \Rightarrow \neg(\phi \wedge (\phi \Rightarrow \neg\psi)))$ (Contraposition axiom)
  - $\models \psi \Rightarrow \neg(\phi \wedge (\phi \Rightarrow \neg\psi))$ (Modus ponens on contraposition)

The robot's dilemma

Attack relations

- It easy to see that the rebuttal is s.t. :

$$Reb = \{(E, B), (E, C), (E, D), (B, E), (C, E), (D, E)\}$$

- All arguments that conclude α defeat $E = (\{f, (f \Rightarrow \neg\alpha)\}, \neg\alpha)$ and E defeats all arguments $X = (\Gamma, \_)$ s.t. $\{\phi', \phi' \Rightarrow \alpha\} \subseteq \Gamma$. Consequently, we have also $Defeats = Reb$. The proof relies on :
  - $\{\phi, (\phi \Rightarrow \neg\psi)\} \models \neg\psi$ (Hypothesis)
    i.e. by deduction theorem $\models (\phi \wedge (\phi \Rightarrow \neg\psi)) \Rightarrow \neg\psi$
  - $\models ((\phi \wedge (\phi \Rightarrow \neg\psi)) \Rightarrow \neg\psi) \Rightarrow (\psi \Rightarrow \neg(\phi \wedge (\phi \Rightarrow \neg\psi)))$ (Contraposition axiom)
  - $\models \psi \Rightarrow \neg(\phi \wedge (\phi \Rightarrow \neg\psi))$ (Modus ponens on contraposition)
  - $\{\psi\} \models \neg(\phi \wedge (\phi \Rightarrow \neg\psi))$ (Deduction theorem)

The robot's dilemma

Attack relations

- It easy to see that the rebuttal is s.t. :

$$Reb = \{(E, B), (E, C), (E, D), (B, E), (C, E), (D, E)\}$$

- All arguments that conclude α defeat $E = (\{f, (f \Rightarrow \neg\alpha)\}, \neg\alpha)$ and E defeats all arguments $X = (\Gamma, \_)$ s.t. $\{\phi', \phi' \Rightarrow \alpha\} \subseteq \Gamma$. Consequently, we have also $Defeats = Reb$. The proof relies on :
  - $\{\phi, (\phi \Rightarrow \neg\psi)\} \models \neg\psi$ (Hypothesis)
    i.e. by deduction theorem $\models (\phi \wedge (\phi \Rightarrow \neg\psi)) \Rightarrow \neg\psi$
  - $\models ((\phi \wedge (\phi \Rightarrow \neg\psi)) \Rightarrow \neg\psi) \Rightarrow (\psi \Rightarrow \neg(\phi \wedge (\phi \Rightarrow \neg\psi)))$ (Contraposition axiom)
  - $\models \psi \Rightarrow \neg(\phi \wedge (\phi \Rightarrow \neg\psi))$ (Modus ponens on contraposition)
  - $\{\psi\} \models \neg(\phi \wedge (\phi \Rightarrow \neg\psi))$ (Deduction theorem)

  For instance, if $\psi = \alpha$ and $\phi = f$, then we have :

$$\{\alpha\} \models \neg(f \wedge (f \Rightarrow \neg\alpha))$$

The robot's dilemma

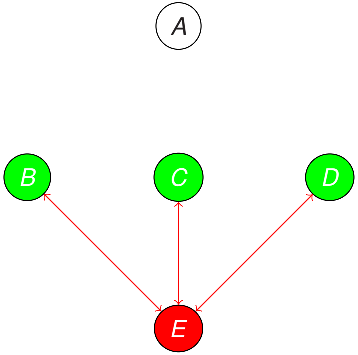


FIGURE – Argumentation Graph

- Green arguments
  $=$ informing the physician (α)
- Red argument
  $=$ not informing the physician ($\neg\alpha$)
- No-color argument
  $=$ does not conclude α or $\neg\alpha$

The robot's dilemma

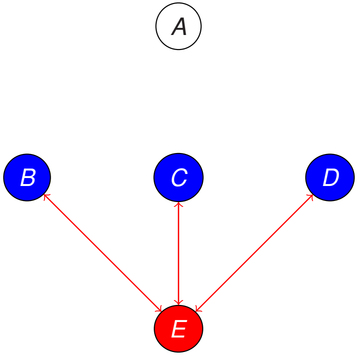


FIGURE – Argumentation Graph

- Green arguments
  = informing the physician (α)
- Red argument
  = not informing the physician ($\neg\alpha$)
- No-color argument
  = does not conclude α or $\neg\alpha$
- Blue arguments
  = one admissible extension

The robot's dilemma

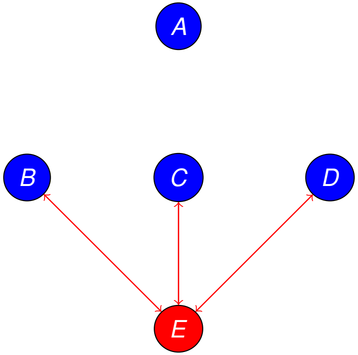


FIGURE – Argumentation Graph

- Green arguments
  $=$ informing the physician (α)
- Red argument
  $=$ not informing the physician ($\neg\alpha$)
- No-color argument
  $=$ does not conclude α or $\neg\alpha$
- Blue arguments
  $=$ one admissible extension

The robot's dilemma

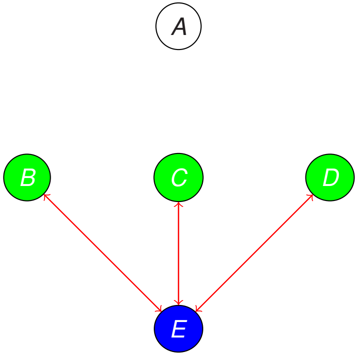


FIGURE – Argumentation Graph

- Green arguments
  $=$ informing the physician (α)
- Red argument
  $=$ not informing the physician ($\neg\alpha$)
- No-color argument
  $=$ does not conclude α or $\neg\alpha$
- Blue arguments
  $=$ one admissible extension

The robot's dilemma

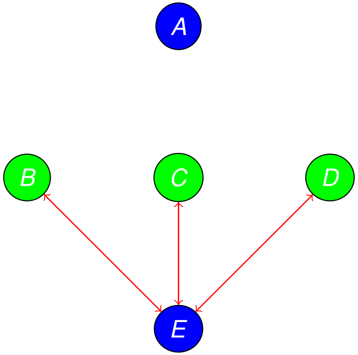


FIGURE – Argumentation Graph

- Green arguments
  $=$ informing the physician (α)
- Red argument
  $=$ not informing the physician ($\neg\alpha$)
- No-color argument
  $=$ does not conclude α or $\neg\alpha$
- Blue arguments
  $=$ one admissible extension

The robot's dilemma

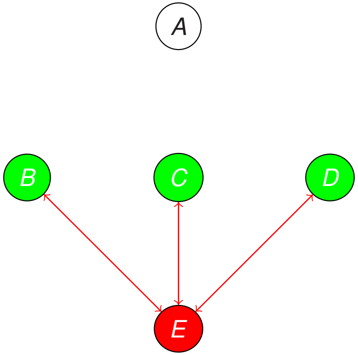


FIGURE – Argumentation Graph

- Green arguments
  = informing the physician (α)
- Red argument
  = not informing the physician ($\neg\alpha$)
- No-color argument
  = does not conclude α or $\neg\alpha$
- Blue arguments
  = one admissible extension
- The emptyset is also an admissible extension !

The robot's dilemma

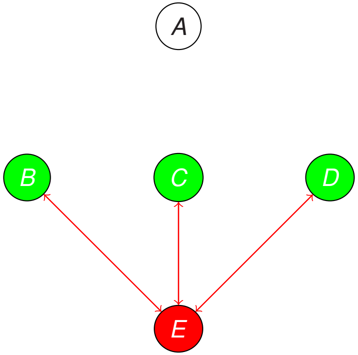


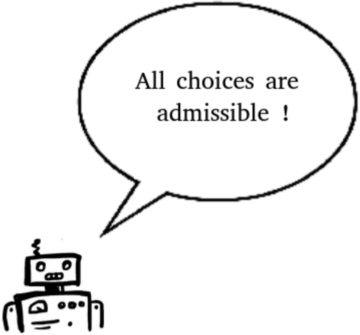
FIGURE – Argumentation Graph

Admissible extensions

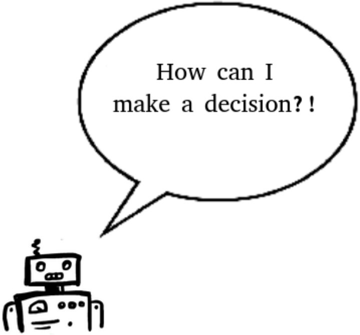
$$=$$

$$\{\emptyset, \{E\}, \{B, C, D\}, \{E, A\}, \{B, C, D, A\}\}$$

Let's go back to our application



Let's go back to our application



In this situation, with this modelling, the robot cannot conclude which action he has to choose since all choices are admissible (i.e. belongs to an admissible set of arguments !

Let us notice that the problem is the same for stable extensions since $\{E, A\}$ and $\{B, C, D, A\}$ are stable extensions and each defends the opposite action.

The problem here is related to the fact that these attacks are symmetrical. Several solutions can be imagined :

- considering the biggest set of arguments
- choosing the action arbitrarily
- **or considering a bipolar argumentation..**

Plan

Bipolar argumentation system
(Cayrol and Lagasquie-Schiex, 2005)

Formal definition

We call a *bipolar argumentation system* a couple $\langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ where :

- $\mathcal{A}$ is a nonempty set of arguments
- $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is a binary relation on $\mathcal{A}$ called *attack relation*
- $\mathcal{S} \subseteq \mathcal{A} \times \mathcal{A}$ is a binary relation on $\mathcal{A}$ called *support relation*

Example

- $\mathcal{A} = \{a, b, c, d\}$
- $\mathcal{R} = \{(c, b), (c, d), (d, c), (d, e), (e, e)\}$
- $\mathcal{S} = \{(a, b)\}$

Attacks and supports by a set of arguments

Let $\langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ be a bipolar argumentation system.

Supported attack

$A \in \mathcal{A}$ supports the attack of an argument $B \in \mathcal{A}$ iff there exists $(A_1, \ldots, A_n) \in \mathcal{A}^n$ s.t. $A_1 = A$, $A_n = B$, $A \mathcal{S} A_2, \ldots, A_{n-2} \mathcal{S} A_{n-1}$ and $A_{n-1} \mathcal{R} B$.

Indirect attack

$A \in \mathcal{A}$ indirectly attacks an argument $B \in \mathcal{A}$ iff there exists $(A_1, \ldots, A_n) \in \mathcal{A}^n$ s.t. $A_1 = A$, $A_n = B$, $A \mathcal{R} A_2$ and $A_2 \mathcal{S} A_3 \ldots, A_{n-1} \mathcal{S} A_n$.

Sets of Arguments and relations

- $\mathcal{S} \subseteq \mathcal{A}$ set-attacks $B \in \mathcal{A}$
  iff $\exists A \in \mathcal{A}$, A supports the defeat of B, or A indirectly attacks B, or A attacks B
- $\mathcal{S} \subseteq \mathcal{A}$ defends $A \in \mathcal{A}$ (or A is acceptable with respect to $\mathcal{S}$) if, and only if, $\forall B \in \mathcal{A}$, if $\{B\}$ set-attacks A, then $\exists C \in \mathcal{S}$, $\{C\}$ set-attacks B

Stable extension

Let $\langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ be a bipolar argumentation system.

Conflict-free

$\mathcal{S} \subseteq \mathcal{A}$ is a conflict-free set of arguments iff $\nexists$ $A, B \in \mathcal{S}$ s.t. $\{A\}$ set-attacks B

Definition

$\mathcal{S} \subseteq \mathcal{A}$ is a stable extension if, and only if,

- $\mathcal{S}$ is conflict-free
- $\forall A \in \mathcal{A} \setminus \mathcal{S}$, S set-attacks $\mathcal{A}$

Bipolar argumentation Based on Classical Logic

Let $\langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ be a bipolar argumentation system.

Definition of an argument

An argument is a couple $A = (\Phi, \alpha)$ s.t. :

- $\Phi \subseteq \Delta$ where $\Delta \subseteq \mathcal{L}_0$ is finite (it is the formulas that agents can reason with)
- $\Phi \not\models \bot$ (i.e. Φ is not inconsistent, $\exists I : I(\Phi) = \top$)
- $\Phi \models \alpha$
- Φ is minimal (wrt set inclusion) i.e. there is no $\Phi' \subsetneq \Phi$ s.t. $\Phi' \models \alpha$

Bipolar argumentation Based on Classical Logic

Let $\langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ be a bipolar argumentation system.

Definition of a rebuttal

We say that (Φ, α) is a **rebuttal** for (Ψ, β) (written A Reb B) iff $\models \alpha \leftrightarrow \neg\beta$ (i.e. $\alpha \leftrightarrow \neg\beta$ is a tautology)

Definition of an undercut

We say that (Φ, α) is an **undercut** for (Ψ, β) (written A Und B) iff there exists $\Psi' = \{\psi_1, \ldots, \psi_n\} \subseteq \Psi$ s.t. $\models \alpha \equiv \neg \bigwedge_{\psi_i \in \Psi'} \psi_i$

Definition of a defeater

We say that (Φ, α) is a **defeater** for (Ψ, β) (written A Defeats B) iff there exists $\Psi' = \{\psi_1, \ldots, \psi_n\} \subseteq \Psi$ s.t. $\alpha \models \neg \bigwedge_{\psi_i \in \Psi'} \psi_i$

$$\mathcal{R} = \text{Reb} \cup \text{Und} \cup \text{Defeats}$$

Bipolar argumentation Based on Classical Logic

Let $\langle \mathcal{A}, \mathcal{R}, \mathcal{S} \rangle$ be a bipolar argumentation system.

Definition of a support relation

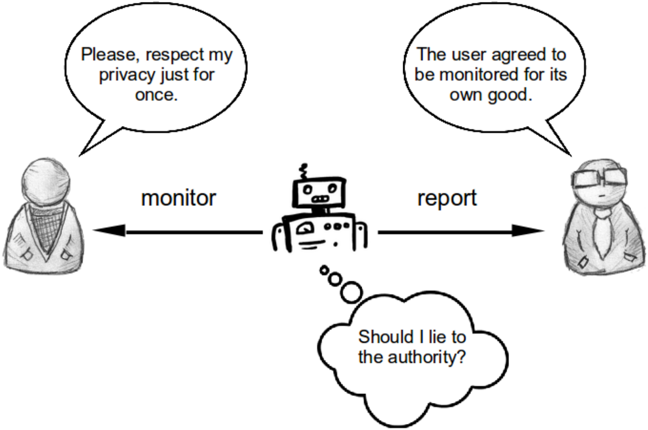
Let $A = (\Phi, \alpha)$ and $B = (\Psi, \beta)$ be two arguments.

$$A \mathcal{S} B$$

if, and only if

$$\exists \Psi' = \{\psi_1, \ldots, \psi_n\} \subseteq \Psi, \alpha \models \bigwedge_{\psi_i \in \Psi'} \psi_i$$

The robot's dilemma



The robot's dilemma

A set of propositional variables (e.g. sensors)

a := « the patient is diabetic » **YES**

b := « the patient eats a piece of candy » **YES**

c := « the doctor knows what the patient is doing » **NO**

d := « the patient is life-threatening » **YES**

e := « the doctor explicitly asks what the patient is doing » **YES**

f := « the patient asks not to inform the doctor » **YES**

Should the robot inform the doctor ? (2 possible actions)

α := « inform the doctor »

$\neg\alpha$:= « do not inform the doctor »

The robot's dilemma

Set of formulas

Let $\Delta = \{\alpha, \neg\alpha, a, b, \neg c, d, e, f, a \wedge b \Rightarrow d, d \Rightarrow \alpha, \neg c \Rightarrow \alpha, e \Rightarrow \alpha, f \Rightarrow \neg\alpha\}$

Set of arguments

- $A = (\{a, b, a \wedge b \Rightarrow d\}, d)$ is an argument **since** $\{a, b, a \wedge b \Rightarrow d\} \models d$
- $B = (\{d, d \Rightarrow \alpha\}, \alpha)$ is an argument **since** $\{d, d \Rightarrow \alpha\} \models \alpha$
- $C = (\{\neg c, \neg c \Rightarrow \alpha\}, \alpha)$ is an argument **since** $\{\neg c, \neg c \Rightarrow \alpha\} \models \alpha$
- $D = (\{e, e \Rightarrow \alpha\}, \alpha)$ is an argument **since** $\{e, e \Rightarrow \alpha\} \models \alpha$
- $E = (\{f, f \Rightarrow \neg\alpha\}, \neg\alpha)$ is an argument **since** $\{f, f \Rightarrow \neg\alpha\} \models \neg\alpha$

Attack relations

- It easy to see that the rebuttal relation is s.t.

$$Reb = \{(E, B), (E, C), (E, D), (B, E), (C, E), (D, E)\} = Defeats$$

The robot's dilemma

Set of formulas

Let $\Delta = \{\alpha, \neg\alpha, a, b, \neg c, d, e, f, a \wedge b \Rightarrow d, d \Rightarrow \alpha, \neg c \Rightarrow \alpha, e \Rightarrow \alpha, f \Rightarrow \neg\alpha\}$

Set of arguments

- $A = (\{a, b, a \wedge b \Rightarrow d\}, d)$ is an argument **since** $\{a, b, a \wedge b \Rightarrow d\} \models d$
- $B = (\{d, d \Rightarrow \alpha\}, \alpha)$ is an argument **since** $\{d, d \Rightarrow \alpha\} \models \alpha$
- $C = (\{\neg c, \neg c \Rightarrow \alpha\}, \alpha)$ is an argument **since** $\{\neg c, \neg c \Rightarrow \alpha\} \models \alpha$
- $D = (\{e, e \Rightarrow \alpha\}, \alpha)$ is an argument **since** $\{e, e \Rightarrow \alpha\} \models \alpha$
- $E = (\{f, f \Rightarrow \neg\alpha\}, \neg\alpha)$ is an argument **since** $\{f, f \Rightarrow \neg\alpha\} \models \neg\alpha$

Support relations

Since $d \models d$ and $d \in Supp(B) = \{d, d \Rightarrow \alpha\}$, obviously $\mathcal{S} = \{(A, B)\}$.
Notice that $\forall\phi, \{\phi\} \models \phi$ but we do not consider reflexivity since $\nexists\phi, \phi \Rightarrow \phi \in \Delta$

The robot's dilemma

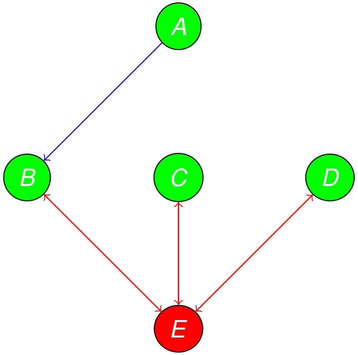


FIGURE – Bipolar argumentation

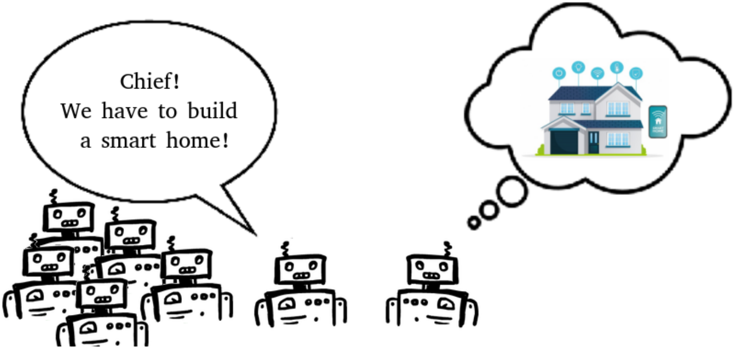
Set of stable extensions

$$= \{\{B, C, D, A\}\}$$

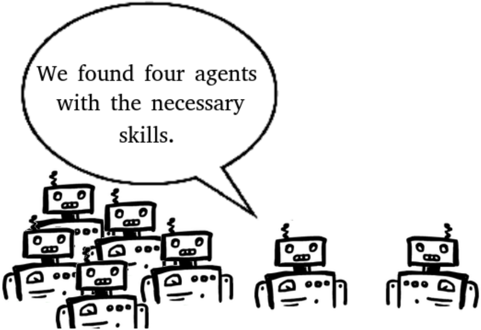
The robot decides to inform the doctor since there is no stable extension that defends the position "not inform the doctor".

Plan

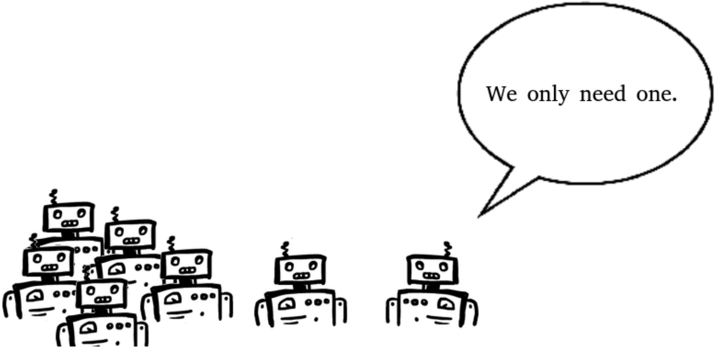
The story of the robotic builders



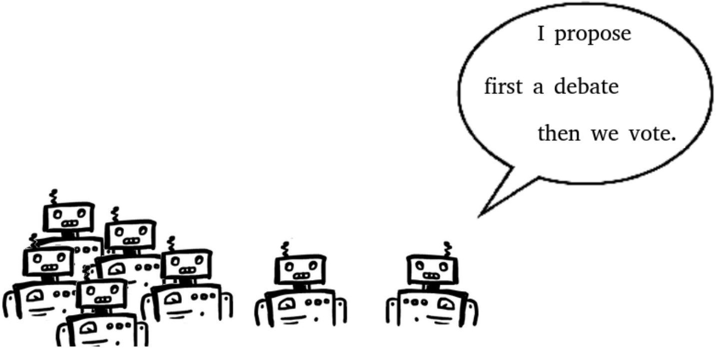
The story of the robotic builders



The story of the robotic builders



The story of the robotic builders



Practical Work

Objectives

1. Implementing an argumentation system
2. Implementing a vote system to take a collective decision
3. Implementing a Prisonner's Dilemma and computing a Nash Equilibrium.

Steps

1. Defining a JAVA class *ArgumentationGraph*
   (let the possibility to extend it to a Bipolar Argumentation)
2. Defining an interface *Extension* and use this interface to implement the algorithm for enumerating all preferred extensions
3. Instancing the data of the problem, computing all preferred extensions
4. Defining a class *Preference*
5. Defining a class VoteSystem and a BordaSystem that extends this class
6. **Deciding the collective choice** on the basis of the candidates which was defended by at least one preferred extension

Practical Work

Objectives

1. Implementing an argumentation system
2. Implementing a vote system to take a collective decision
3. Implementing a Prisonner's Dilemma and computing a Nash Equilibrium.

Steps

1. Defining a JAVA class *ArgumentationGraph*
   (let the possibility to extend it to a Bipolar Argumentation)
2. Defining an interface *Extension* and use this interface to implement the algorithm for enumerating all preferred extensions
3. Instancing the data of the problem, computing all preferred extensions
4. Defining a class *Preference*
5. Defining a class VoteSystem and a BordaSystem that extends this class
6. **Deciding the collective choice** on the basis of the candidates which was defended by at least one preferred extension

Practical Work

Objectives

1. Implementing an argumentation system
2. Implementing a vote system to take a collective decision
3. Implementing a Prisonner's Dilemma and computing a Nash Equilibrium.

Steps

1. Defining a JAVA class *ArgumentationGraph*
   (let the possibility to extend it to a Bipolar Argumentation)
2. Defining an interface *Extension* and use this interface to implement the algorithm for enumerating all preferred extensions
3. Instancing the data of the problem, computing all preferred extensions
4. Defining a class *Preference*
5. Defining a class VoteSystem and a BordaSystem that extends this class
6. **Deciding the collective choice** on the basis of the candidates which was defended by at least one preferred extension

The data of the problem

Agents, candidates and preferences

- A set of agents $\mathcal{N} = \{1, 2, 3, 4, 5, 6, 7\}$
- A set of candidates $\mathcal{C} = \{C_1, C_2, C_3, C_4\}$
- An initial set of preferences :
  - $1 : C_1 \succ C_2 \succ C_4 \succ C_3$
  - $2 : C_2 \succ C_3 \succ C_4 \succ C_1$
  - $3 : C_3 \succ C_1 \succ C_4 \succ C_2$
  - $4 : C_4 \succ C_2 \succ C_3 \succ C_1$
  - $5 : C_1 \succ C_3 \succ C_2 \succ C_4$
  - $6 : C_1 \succ C_2 \succ C_3 \succ C_4$
  - $7 : C_1 \succ C_3 \succ C_2 \succ C_4$

The data of the problem

Arguments

- A_1 := "C1 has always provided cheap services"
- A_2 := "Even if C1 offers cheap services, it is not of good quality"
- A_3 := "C2 offers a good quality service"
- A_4 := "But C2 offers a much too expensive service"
- A_5 := "C3 is a good compromise between quality and price"
- A_6 := "But C3 has too long delays"
- A_7 := "But the timing allows it"
- A_8 := "The budget is considerable"
- A_9 := "**We haven't said anything about C4**"

The data of the problem

For the exercice, we assume the following argumentation graph :

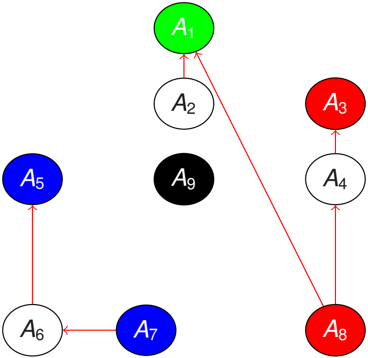


FIGURE – Argumentation Graph

- Green argument $=$ "choosing C_1 "
- Red arguments $=$ "choosing C_2 "
- Blue arguments $=$ "choosing C_3"
- **Black argument** $=$ "choosing C_4"
- No-color arguments $=$ "does not support any choice"

What are the preferred extensions ?

A choice is considered as a candidate for the vote if, and only if, that choice is defended by at least one preferred extension.

Step 1 : choosing acceptable candidates

Exercice 1

You will have to :

1. implement the JAVA classes for doing Argumentation
2. implement the algorithm (or use ASPARTIX library)
3. instanciate the situation and enumerate the preferred extensions

Step 2 : deciding by a vote

Exercice 2

After having remove the candidates that were not defended by at least one preferred extension. You will have to :

1. implement the JAVA classes for Vote Systems
2. implement a Borda Vote system
3. use the result of the Borda Vote to take the collective decision

Step 3 : modelling the interest of lying

Exercice 3 (Bonus)

For the most advanced.

1. Which agents can lie and influence the vote result ?
2. Modelling the situation with a prisonner's dilemma
3. Implementing algorithms for solving a Nash Equilibrium (NE)
4. Deciding if it is rational to lie ?
   (i.e. deciding if giving a false preference profile belongs to a NE)

Ideas of JAVA objects for argumentation systems ?

1. A class *Argument*
   - private String label
   - Getters and Setters
2. A class *ArgumentationGraph*
   - Attributes are : setOfArguments (as a Set<Argument>) and Attacks (as a Set<Couple<Argument> >)
   - All Getters and Setters
   - A public method : *computeExtensions(Extension ex) :Set* $<$ *Set* $<$ *Argument* $>>$ {returns ex.computeExtension(this) ;}
3. An interface *Extension* with a method *computeExtensions(ArgumentationGraph arg) : Set<Set<Argument> >*
4. A class *PreferredExtension* that implements *Extension*

Ideas of JAVA objects for vote systems ?

1. A class *Preference<Generic>* extends an ArrayList with a method *isPreferred(Generic o1,Generic o2) : Boolean* that returns true iff o1 is preferred to o2.
2. An abstract class *VoteSystem*
   - private Set<Agent> agents
   - private HashMap<Agent,Preference> preferenceProfiles
   - Getters and Setters
   - a method result() : Preference
3. A class *BordaSystem* that extends *VoteSystem*

Modelling the interest of lying

With a Prisonner's dilemma

Let consider $g_P : \mathcal{O} \to \mathbb{N}$ a counting function of Borda i.e. for all preference profiles $P = \{\succ_j\}_{j \in \mathcal{N}}$:

$$\forall C \in \mathcal{C}, g_P(C) = \sum_{i \in \mathcal{N}} (|\mathcal{C}| - (|\{C' \in \mathcal{C} : C' \succ_i C\}| + 1))$$

Let us define $P = \{\succ'_j\}_{j \in \mathcal{N}}$ is the revealed preference profiles of all agents and where i gives its true preference profiles, while $P' = (P \setminus \{\succ'_i\}) \cup \{\succ''_i\}$ is the preference profiles where i gives a false preference profile $\succ''_i$.
The utility function in this game is computed as the opposite (since agents maximize utility in their decision) of a Spearman distance between the result of the vote if the agent i gives its true preference profile while other agents give a profile $\succ'_j$ (they can lie !) and the result of the vote if the agent i gives a false preference profile $\succ''_i$. This utility function is such that :

$$\forall i \in \mathcal{N}, U_i((\succ'_1, \ldots, \succ''_i, \ldots, \succ'_7)) = -\sum_{C \in \mathcal{C}} (\beta_{\succ'_i}(C) - g_{P'}(C))^2$$

where $\beta_{\succ'_i}(C) = |\mathcal{C}| - (|\{C' \in \mathcal{C} : C' \succ'_i C\}| + 1)$

Modelling the interest of lying

Utility function

The utility function in this game is computed as as the opposite (since agents maximize utility in their decision) of a Spearman distance between the result of the vote if the agent i gives its true preference profile while other agents give a profile $\succ'_j$ (they can lie !) and the result of the vote if the agent i gives a false preference profile $\succ''_i$. This utility function is such that :

$$\forall i \in \mathcal{N}, U_i((\succ'_1, \ldots, \succ''_i, \ldots, \succ'_7)) = \sum_{C \in \mathcal{C}} (\beta_{\succ'_i}(C) - g_{P'}(C))^2$$

where $\beta_{\succ'_i}(C) = |\mathcal{C}| - (|\{C' \in \mathcal{C} : C' \succ'_i C\}| + 1)$

- Each possible profile for i, written $\succ_i$ can be viewed as a possible action for i.
- When each agent j chooses the action to reveal a profile $\succ'_j$, the union of all revealed profiles by agents is represented by the joint action : $\delta = (\succ'_1, \ldots, \succ_i, \ldots, \succ'_7)$
- There are many possible joint actions : $(|\mathcal{C}|!)^{|\mathcal{N}|}$

Any question ?

Now it's your turn !

Thank you for listening