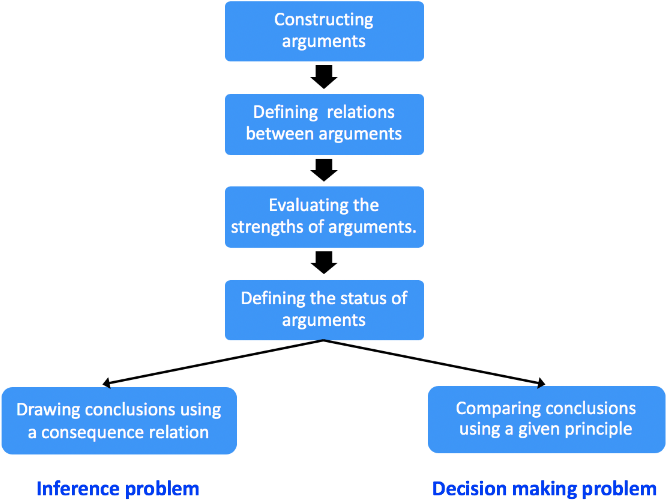
Arguing in Multi-Agent System II

Wassila Ouerdane & Nicolas Sabouret

CentraleSupélec

28 février 2019

Previously : an argumentation process



Previously : Step 3 (Status of arguments ?)

Two ways for computing the status of arguments :

- The declarative form usually requires fixed point definition, and establishes certain sets of arguments as acceptable $\rightsquigarrow$ Acceptability semantics.
- The procedural form amounts to defining a procedure for testing whether a given argument is a member of “a set of acceptable arguments” $\rightsquigarrow$ Proof theory.

Proof theory for abstract argumentation

- Argument games between proponent P and opponent O:
- Proponent starts with an argument
- Then each party replies with a suitable attacker
- A winning criterion (e.g. the other player cannot move)
- Acceptability status corresponds to existence of a winning strategy.

[S. Modgil, Martin W.A. Caminada, 2009]. *Proof Theories and Algorithms for Abstract Argumentation Frameworks.Chapter in Argumentation in Artificial Intelligence (I. Rahwan, G. Simari, eds.), Springer Publishing Company, Incorporated, 2009.*

Proof theories

- Let $\langle A, R \rangle$ an argumentation system
- $\mathcal{E}_1, \ldots, \mathcal{E}_n$ its extension under a given semantics.

Problem

Let $a \in Args$

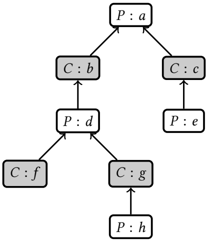
- Is a in one extension ?
- Is a in every extension ?

Proof theories [Amgoud and Cayrol, 2000]

A dialogue is a non-empty sequences of moves such that :

$$Move_i = (Player_i, Arg_i)(i \geq 0)$$

- $Player_i = P$ iff i is even, $Player_i = C$ iff i is odd
- $Player_0 = P$ and $Arg_0 = a$
- $Player_i = Player_j = P$ and $i \neq j$ then $Arg_i \neq Arg_j$
- if $Player_i = P$ $(i > 1)$ then Arg_i attacks Arg_{i-1}



Proof theories

A dialogue tree is a finite tree where each branch is a dialogue [Amgoud and Caryrol, 2000].

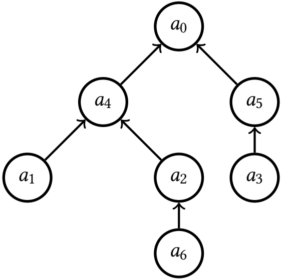


Figure – $\langle A, R\rangle$

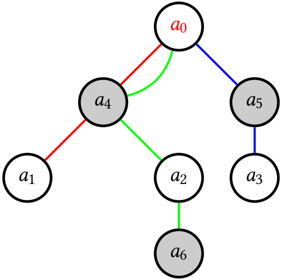
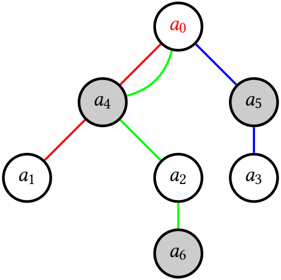


Figure – Dialogue Tree

Proof theories

A player wins a dialogue iff it ends the dialogue



——– won by P
——– won by P
——– won by C

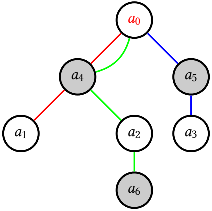
Proof theories

- A candidate sub-tree is a sub-tree of the dialogue tree containing *all* the edges of an even move (P) and *exactly one edge* of an odd move (C).
- A solution sub-tree is a candidate subtree whose branches are all won by P
- P wins a dialogue tree iff the dialogue tree has a solution sub-tree.

Complete construction :
$a \in$ the grounded extension iff $\exists$ a dialogue tree whose root is a and won by P

Note : For any argumentation framework, there is guaranteed to be exactly one grounded extension

Example-1



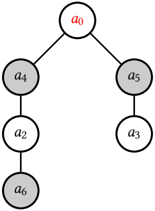
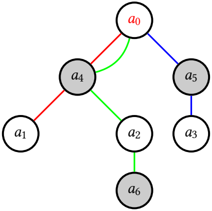


Figure – Sub-Tree S_1

Proof Theories

Example-1



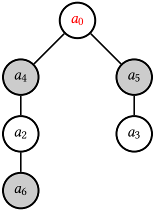


Figure – Sub-Tree S_1

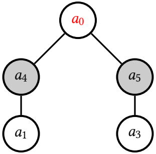
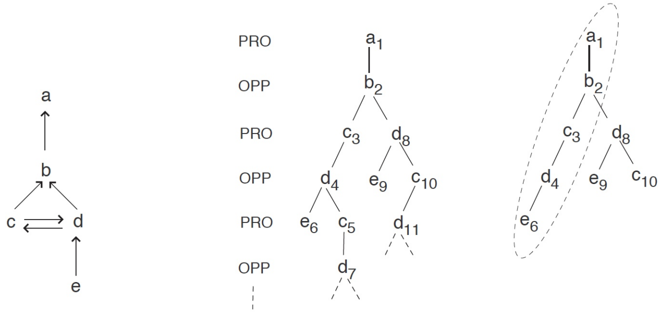


Figure – Sub-Tree S_2

Each branch of S_2 is won by $P \to S_2$ is a solution sub-tree
a_0 is in the grounded extension $\{a_0, a_1, a_3\}$.

Proof Theories



i) ii) iii)

Example-2

- An argumentation framework in i).
- Part of the dispute tree induced by a is shown in ii), and
- the dialogue tree induced by a is shown in Figure iii).
- Observe that $\{(a_1, b_2, c_3, d_4, e_6)\}$ is a-winning strategy for a (the grounded extension {a, c, e})

Dialogue types [Walton and Krabbe, 1995]

Types of dialgue	Initial situation	Participant's goal	Goal of dialogue
Persuasion	Conflicts of opinions	Persuade other party	Resolve or clarify issue
Negotiation	Conflicts of interests	Get what you most want	Reasonable settlement
Information seeking	One party lacks information	Acquire or give information	Exchange Information
Deliberation	Dilemma or practical choice	Coordinate goals or actions	Decide best course of action
...			

Components of a dialogue System

- Communication language + Domain Language
- Protocol : the set of rules for generating coherent dialogues.
- Agent strategies : the set of tactics used by the agents to choose a move to play.
- Outcome :
  - one of a set of possible deals or
  - conflicts

Protocol+ Strategies $\rightarrow$ Outcome

Components of a dialogue System

Communication language

- A syntax : a set of locutions, utterances or speech acts (propose, argue, accept, reject, etc.)
- A semantic : a unique meaning for each utterance (a locution gives the same content a different meaning).

Components of a dialogue System

Communication language

- A syntax : a set of locutions, utterances or speech acts (propose, argue, accept, reject, etc.)
- A semantic : a unique meaning for each utterance (a locution gives the same content a different meaning).

Domain language

Concepts about the environment, the different agents, the time, proposals, etc.

Example [Sierra et al., 1998]

- `offer` (a, b, price=200$ and Item=palm130, t_1) : means the agent a proposes to agent b at time t_1 the sale of item palm130 for the price 200$.
- `reject` (b, a, price=200$ and item =palm130, t_2) : means that agent b rejects such a proposal made by agent a.

Dialogue protocols

Main parameters :

- the set of allowed moves (e.g. Claim, Argue, ...)
- the possible replies for each move
- the number of moves per turn
- the turntaking
- the notion of Backtracking
- the computation of the outcome

Note : the set of parameters identifies more or less rich dialogue.

Dialogue protocols

Computing the outcome

Two approaches

- the protocol is equipped with an argumentation system that evaluates the content
- the rules of the proof theory are encoded in the protocol

A negotiation dialogue

Example

Buyer :	Can't you give me this 806 a bit cheaper ?
Seller :	Sorry that's the best I can do. Why don't you go for a Polo instead ?
Buyer :	I have a big family and I need a big car (B1)
Seller :	Modern Polo are becoming very spacious and would easily fit in a big family. (S1)
Buyer :	I didn't know that, let's also look at Polo then

Negotiation dialogue

- Negotiation is essential in settings where autonomous agents have conflicting interests and a desire to cooperate
- Various interaction and decision mechanisms for automated negotiation have been proposed and studied :
  - game-theoretic analysis (Rosenschein & Zlotkin, 1994 ; Kraus, 2001 ; Sandholm, 2002) ;
  - heuristic-based approaches (Faratin, 2000 ; Kowalczyk & Bui, 2001 ; Fatima et al., 2002) ; and
  - argumentation-based approaches (Kraus et al., 1998 ; Parsons et al., 1998 ; Sierra et al., 1998)

Negotiation protocol

- The interaction protocol specifies, at each stage of the negotiation process, who is allowed to say what.
- For example, after one agent makes a proposal, the other agent may be able to accept it, reject it or criticize it, but might not be allowed to ignore it by making a counterproposal
- The protocol might be based solely on the last utterance made, or might depend on a more complex history of messages between agents.

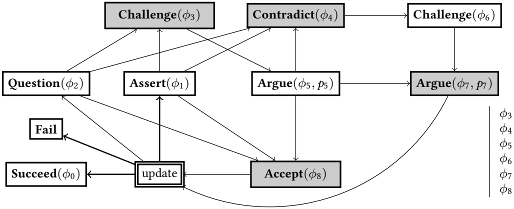
Negotiation protocol : some components

- rules for admission, which specify when an agent can participate in a negotiation dialogue and under what conditions ;
- termination rules, which specify when an encounter must end (e.g., if one agent utters an acceptance locution) ;
- rules for proposal validity, which specify when a proposal is compliant with some conditions (e.g., an agent may not be allowed to make a proposal that has already been rejected) ;
- rules for outcome determination, which specify the outcome of the interaction ; in argumentation-based frameworks, these rules might enforce some outcome based on the underlying theory of argumentation (e.g., if an agent cannot construct an argument against a request, it accepts it [Parsons et al., 1998]) ;

Negotiation Protocol

How to specify an interaction protocol ?

- Finite-state machines [Parsons et al., 1998] : explicit specification of interaction protocoles.
  ⇝ useful when the interaction involves a limited number of locutions



$\phi_3 \subseteq \phi_1$, $\phi_3 \subseteq \phi_2$
$\phi_4 \subseteq \phi_1$, $\phi_4 \subseteq \phi_2$, $\phi_4 \subseteq \phi_5$
$\phi_5 \subseteq \phi_3$
$\phi_6 \subseteq \phi_4$
$\phi_7 \subseteq \phi_6$, $\phi_7 \subseteq \phi_5$ or ϕ_7 conflicting with ϕ_5
$\phi_8 \subseteq \phi_1$, $\phi_8 \subseteq \phi_2$, $\phi_8 \subseteq \phi_5$

Ch. Labreuche, N. Maudet, W. Ouerdane, and S. Parsons. A dialogue game for recommendation with adaptive preference models. International Conference on Autonomous Agents and Multiagent systems, May 2015, Istanbul, Turkey. pp.959-967.

Negotiation Protocol

How to specify an interaction protocol ?

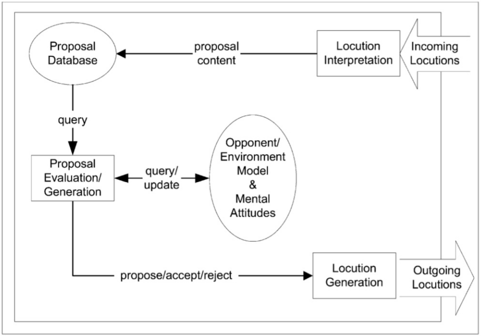
- Finite-state machines [Parsons et al., 1998] : explicit specification of interaction protocoles.
  $\rightsquigarrow$ Advantage : useful when the interaction involves a limited number of locutions
- Dialogue games [McBurney et al., 2003], by stating clearly the pre and post conditions of each locution as well as its effects on agent's commitments.
  $\rightsquigarrow$ Advantage : providing clear and precise semantics of the dialogue

Negotiation Protocol

This locution allows a seller (or advisor) agent to announce that it (or another seller) is willing to sell a particular option [McBurney et al. ,2003].

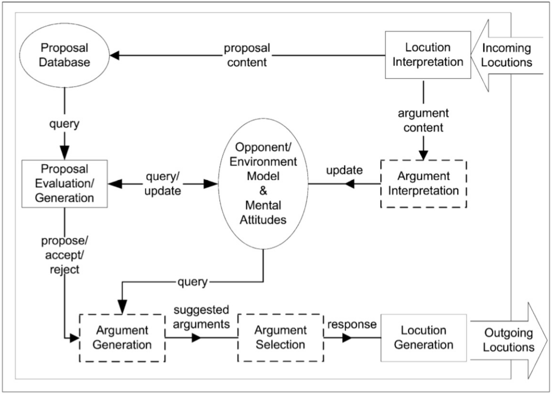
- **Locution** : `willing_to_sell` (P_1, T, P_2, V), where P_1 is either an advisor or a seller, T is the set of participants, P_2 is a seller and V is a set of sales options.
- **Preconditions** : some participant P_3 must have previously uttered a locution `seek_info` (P_3, S, p) where $P_1 \in S$ (the set of sellers), and the options in V satisfy constraint p.
- **Meaning** : the speaker P_1 indicates to audience T that agent P_2 is willing to supply the finite set $V = \{a, b, ...\}$ of purchase options to any buyer in set T. Each of these options satisfy constraint p uttered as part of the prior `seek(.)` locution.
- Response : none required.
- **Information store updates** : for each $\neg a \in V$, the 3-tuple $(T, P_2, \neg a)$ is inserted into $IS(P_1)$, the information store for agent P_1.
- Commitment store updates : no effects.

Conceptual elements of a classical negotiating agent



Borrowed from : https ://eprints.soton.ac.uk/258850/1/abn.pdf

Conceptual elements of an Argument Based Negotiation agent

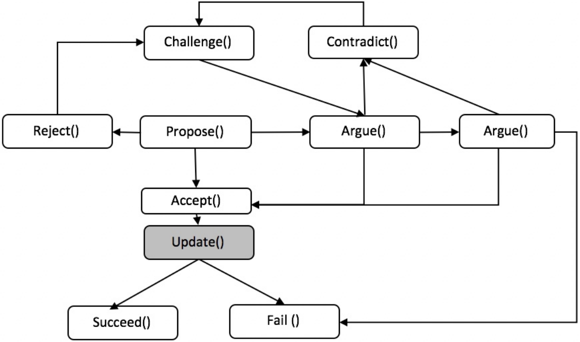


Borrowed from : https ://eprints.soton.ac.uk/258850/1/abn.pdf

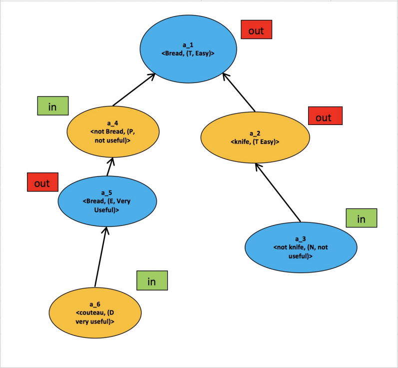
Lost Island problem

$Agent_i$	Turn	Arg	Locution
$Agent_1$	1		Propose (Bread)
$Agent_2$	2		Why (Bread) ?
$Agent_1$	3	a_1	Argue(none, Bread, (transport, easy))
$Agent_2$	6	a_2	Argue(a_1, knife, (transport, easy))
$Agent_1$	7		Contradict (a_2, not knife)
$Agent_2$	8		Why (not knife) ?
$Agent_1$	9	a_3	Argue (a_2, not knife, (to eat, not useful), (to eat $\succ$ transport))
$Agent_2$	10		Contradict(a_1, not bread)
$Agent_1$	11		Why (not bread) ?
$Agent_2$	12	a_4	Argue(a_1, not bread, (protection, not useful), (protection>$\succ$ transport))
$Agent_1$	13	a_5	Argue(a_4, Bread, (to eat, very useful), (to eat $\succ$ protection))
$Agent_2$	14	a_6	Argue (a_5, knife, (protection, very useful), protection $\succ$ to eat))
$Agent_1$	15		accept (knife, a_6)
$Agent_1$	16		check(bag)
$Agent_1$	17		commit (knife, bag)
$Agent_2$	18		Commit (knife, bag)
$Agent_2$	19		send_message(Central, add(knife))

Lost Island problem



Lost Island problem



Abstract argumentation system : Labelling

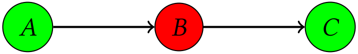
Example of labelling

- a labelling is a total function
$$\mathcal{L} : \texttt{Args} \rightarrow \{ \texttt{in}, \texttt{out}, \texttt{und} \}$$
- an argument x is `in` iff every y that attacks x is `out`
- an argument x is `out` iff there is at least one y that attacks x is `in`
- an argument x is `und` iff not every y that attacks x is labelled `out` , and there is no y that attacks x such that y is labelled `in`

Abstract argumentation system : Labelling

Example of labelling

- a labelling is a total function
$$\mathcal{L} : \texttt{Args} \rightarrow \{ \texttt{in}, \texttt{out}, \texttt{und} \}$$
- an argument x is `in` iff every y that attacks x is `out`
- an argument x is `out` iff there is at least one y that attacks x is `in`
- an argument x is `und` iff not every y that attacks x is labelled `out` , and there is no y that attacks x such that y is labelled `in`



[S. Modgil, Martin W.A. Caminada, 2009]. *Proof Theories and Algorithms for Abstract Argumentation Frameworks.Chapter in Argumentation in Artificial Intelligence (I. Rahwan, G. Simari, eds.), Springer Publishing Company, Incorporated, 2009.*