Computational Models of Dialogical Argumentation

Anthony Hunter

Department of Computer Science, University College London, UK

October 12, 2012

Argumentation as a cognitive process

Argumentation is a key way humans deal with conflicting information

- Argumentation involves identifying arguments and counterarguments relevant to an issue. (e.g. What are the pros and cons for the safety of mobile phones for children?)
- Argumentation involves weighing, comparing, or evaluating arguments. (e.g. What sense can we make of the arguments concerning mobile phones for children?)
- Argumentation may involve drawing conclusions. (e.g. A parent answering the question "Are mobile phones safe for my children?").

Argumentation as a cognitive process

Arguments are normally based on imperfect information

Arguments are normally constructed from information that is incomplete, inconsistent, uncertain and/or subjective, and from multiple heterogeneous sources.

Diverse examples of arguments

Mathematical	All squares have fours corners. That is a square, and so it has four corners.
Epsitemic	If I had a sister, I would know about it. As I don't know about it, I don't have a sister.
Scientific	Mr Jones has angina, therefore prescribe him daily aspirin.
Subjective	This film should have won an Oscar because it was a good movie with an edge.

Counterarguments

Since arguments are normally constructed from imperfect information, there are often counterarguments.

Argumentation as a cognitive process

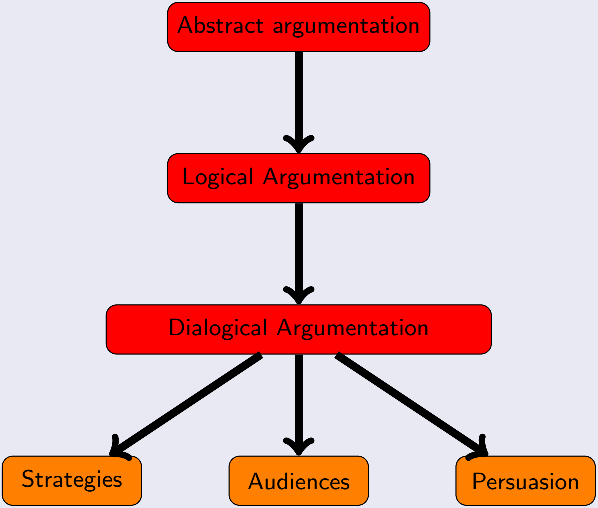
Argumentation may involve convincing an audience

Argumentation as a cognitive process

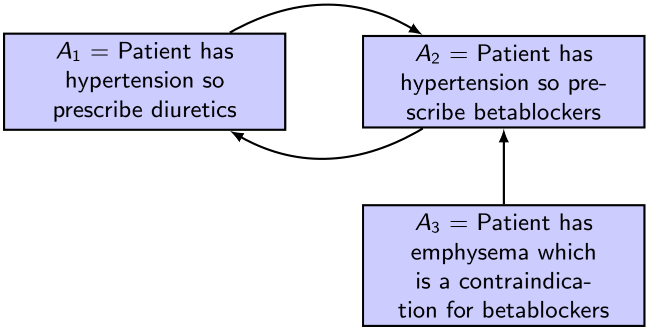
Argumentation may involve exchanges of arguments in a dialogue

Overview of the talk

Levels of complexity in computational models



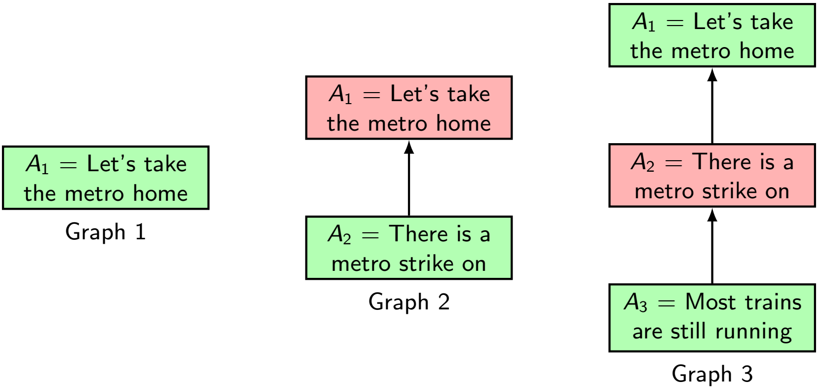
Abstract argumentation: Graphical representation



[See Dung (AIJ 1995)]

Abstract argumentation: Winning arguments

Green means the argument "wins" and red means the argument "looses".

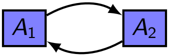


[See Simari+Loui (AIJ 1992); Pollock (AIJ 1995),etc.]

Abstract argumentation: Extensions

Types of extension for a set of arguments

Admissible	iff it is conflictfree and defends all its members
Complete	iff it is admissible and all arguments it defends are in it
Grounded	iff it is minimal (w.r.t set inclusion) complete
Preferred	iff it is maximal (w.r.t set inclusion) complete
Stable	iff it is preferred and attacks all arguments not in it



	admissible	complete	grounded	preferred	stable
$\{\}$	✓	✓	✓		
$\{A_1\}$	✓	✓		✓	✓
$\{A_2\}$	✓	✓		✓	✓
$\{A_1, A_2\}$					

Abstract argumentation: Extensions

Types of extension for a set of arguments

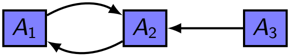
Admissible iff it is conflictfree and defends all its members

Complete iff it is admissible and all arguments it defends are in it

Grounded iff it is minimal (w.r.t set inclusion) complete

Preferred iff it is maximal (w.r.t set inclusion) complete

Stable iff it is preferred and attacks all arguments not in it

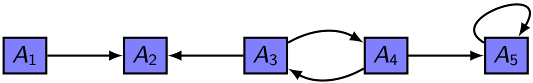


	admissible	complete	grounded	preferred	stable
$\{\}$	✓				
$\{A_1\}$	✓				
$\{A_3\}$	✓				
$\{A_1, A_3\}$	✓	✓	✓	✓	✓

Abstract argumentation: Extensions

Types of extension for a set of arguments

- **Admissible** iff it is conflictfree and defends all its members
- **Complete** iff it is admissible and all arguments it defends are in it
- **Grounded** iff it is minimal (w.r.t set inclusion) complete
- **Preferred** iff it is maximal (w.r.t set inclusion) complete
- **Stable** iff it is preferred and attacks all arguments not in it



	admissible	complete	grounded	preferred	stable
$\{A_1\}$	✓	✓	✓		
$\{A_1, A_3\}$	✓	✓		✓	
$\{A_1, A_4\}$	✓	✓		✓	✓

Abstract argumentation: Conclusions

- Abstract argumentation has formalized the notion of dialectics that is important in argumentation (See Dung AIJ 1995).
- Abstract argumentation has been extended in various ways (e.g. preferences, weights, probabilities, etc.)
- There is some work on using technology from natural language processing for building argument graphs (e.g. information extraction, sentiment analysis, text entailment, etc).
- However, abstract arguments are atomic, and so have no internal structure.
- To better understand, and to generate arguments, we require logical arguments.

Logical argumentation: Arguments

Argument

An **argument** from a set of formulae Δ is a pair $\langle \Phi, \alpha \rangle$ such that

1. $\Phi \subseteq \Delta$
2. $\Phi \nvdash \bot$
3. $\Phi \vdash \alpha$
4. there is no $\Phi' \subset \Phi$ such that $\Phi' \vdash \alpha$.

We call Φ the **support** of the argument and α the **claim** of the argument. The support of an argument is the justification/explanation for the claim.

Example using classical logic

If $\Delta = \{\alpha, \alpha \rightarrow \beta, \beta \rightarrow \gamma, \delta \rightarrow \neg\beta\}$, then arguments from Δ include:

$$\begin{array}{cc} \langle \{\alpha\}, \alpha \rangle & \langle \{\alpha, \alpha \rightarrow \beta\}, \beta \rangle \\ \langle \{\alpha, \alpha \rightarrow \beta, \beta \rightarrow \gamma\}, \gamma \rangle & \langle \{\alpha \rightarrow \beta\}, \alpha \rightarrow \beta \rangle \\ \langle \{\alpha \rightarrow \beta\}, \neg\alpha \vee \beta \rangle & \langle \{\}, \neg\alpha \vee \alpha \rangle \end{array}$$

[See Besnard and Hunter (2008) *Elements of Argumentation*, MIT Press]

Logical argumentation: Attacks by counterarguments

Counterarguments

If $\langle \Phi, \alpha \rangle$ and $\langle \Psi, \beta \rangle$ are arguments, then

- $\langle \Phi, \alpha \rangle$ **rebuts** $\langle \Psi, \beta \rangle$ iff $\alpha \vdash \neg \beta$
- $\langle \Phi, \alpha \rangle$ **undercuts** $\langle \Psi, \beta \rangle$ iff $\alpha \vdash \neg \bigwedge \Psi$

Direct undercut

A **direct undercut** for an argument $\langle \Phi, \alpha \rangle$ is an argument of the form $\langle \Psi, \neg \phi_i \rangle$ where $\phi_i \in \Phi$.

Example using classical logic

$\langle \{\beta, \beta \rightarrow \alpha\}, \alpha \rangle$ rebuts $\langle \{\gamma, \gamma \rightarrow \neg \alpha\}, \neg \alpha \rangle$

$\langle \{\gamma, \gamma \rightarrow \neg \beta\}, \neg(\beta \wedge (\beta \rightarrow \alpha)) \rangle$ undercuts $\langle \{\beta, \beta \rightarrow \alpha\}, \alpha \rangle$

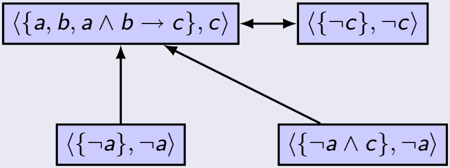
$\langle \{\delta \rightarrow \neg \beta\}, \neg \beta \rangle$ is a direct undercut for $\langle \{\alpha, \beta\}, \alpha \wedge \beta \rangle$

Logical argumentation: Attacks by counterarguments

A rebut denotes a disagreement with the claim, whereas an undercut denotes a disagreement with the support (i.e. a disagreement of the explanation or justification).

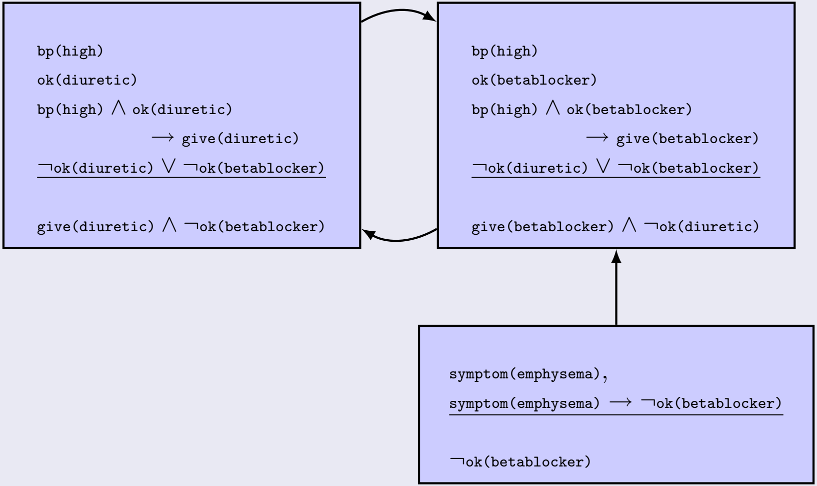
Example

- $a =$ "garlic is horrible"
- $b =$ "this dish contains garlic"
- $c =$ "this dish is horrible"



Logical argumentation

Example using classical logic



Logical argumentation:Conclusions

- Logical argumentation can instantiate abstract argumentation.
- A variety of logics have been considered for argumentation (e.g. defeasible logic, classical logic, temporal logic, probabilistic logic, & non-monotonitc logic).
- A range of frameworks have been developed with implementations
  - Deductive argumentation (Hunter, Besnard, Cayrol, Amgoud, et al)
  - Defeasible logic programming (Simari, et al)
  - Assumption-based argumentation (Toni, et al)
  - ASPIC+ (Prakken, et al)
  - Carneades (Gordon, et al)
- A variety of application areas are being developed (e.g. law, medicine, egovernment, engineering design, & semantic web).

Dialogical argumentation

Discussions where agents collaborate to understand a topic.

Negotiations where agents try to find an agreed solution.

Debates where agents try to persuade each other.

Court cases where advocates try to defeat the opposition case.

Modelling dialogical argumentation

Components of a model of dialogical argumentation

- **Participants** Specification of the information held by each agent (e.g. a knowlegebase, a set of goals, etc.)
- **Moves** Specification of the moves that can be made (e.g. `why`(ϕ), `claim`(ψ), `posit`(A), etc.)
- **Protocol** The rules of the game (i.e. the moves an agent is allowed, or is obliged, to make at each stage of the dialogue).

[See Hamblin (Theoria 1971); MacKenzie (JPL 1979)]

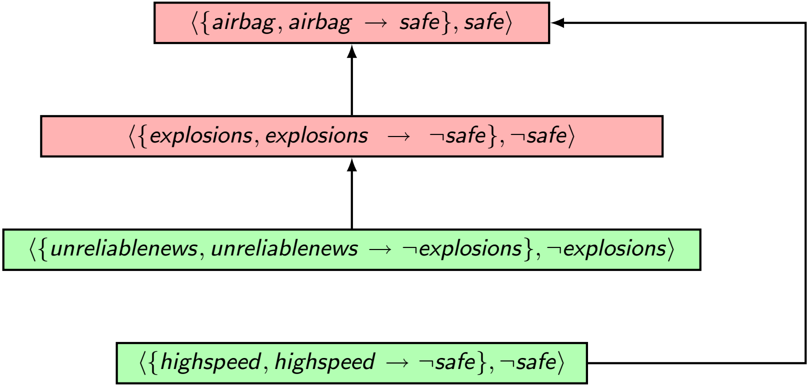
Modelling dialogical argumentation

Dialogue between Paul and Olga about Paul's claim that a particular car is safe.

1. $\texttt{claim}(\texttt{Paul}, \texttt{safe})$
2. $\texttt{why}(\texttt{Olga}, \texttt{safe})$
3. $\texttt{explain}(\texttt{Paul}, \{\texttt{airbag}, \texttt{airbag} \rightarrow \texttt{safe}\}, \texttt{safe})$
4. $\texttt{concede}(\texttt{Olga}, \texttt{airbag})$
5. $\texttt{explain}(\texttt{Olga}, \{\texttt{explosions}, \texttt{explosions} \rightarrow \neg\texttt{safe}\}, \neg\texttt{safe})$
6. $\texttt{explain}(\texttt{Paul}, \{\texttt{unreliablenews}, \texttt{unreliablenews} \rightarrow \neg\texttt{explosions}\}, \neg\texttt{explosions})$
7. $\texttt{explain}(\texttt{Olga}, \{\texttt{highspeed}, \texttt{highspeed} \rightarrow \neg\texttt{safe}\}, \neg\texttt{safe})$

[Example adapted from Prakken (KER 2006)]

Modelling dialogical argumentation



Specifying dialogical argumentation

Specification using executable logic

Each dialogical argumentation system is specified in a standard way where

- a dialogue is a sequence of states
- a set of logical rules is used to specify the allowed transitions

Execution state

Each of the following is a set of literals for each step n of the dialogue

$$
\begin{aligned}
s_1(n) &= \text{private state of agent 1} \\
a_1(n) &= \text{action state of agent 1} \\
p(n) &= \text{public state of agents} \\
a_2(n) &= \text{action state of agent 2} \\
s_2(n) &= \text{private state of agent 2}
\end{aligned}
$$

An execution (i.e. a dialogue)

$(s_1, a_1, p, a_2, s_2, t)$ where t is the index of the terminal state

Specifying dialogical argumentation

The language for literals is unrestricted, and no particular meaning is ascribed to them in general.

Examples of literals in a public or private state

$$
\begin{array}{l}
\mathtt{belief}(\mathtt{a} \wedge \mathtt{b}) \\
\mathtt{belief}(\mathtt{a} \rightarrow \mathtt{c}) \\
\mathtt{issue}(\mathtt{c}) \\
\mathtt{commit}(\mathtt{c}) \\
\mathtt{prefer}(\mathtt{d}, \mathtt{b} \wedge \mathtt{c}) \\
\mathtt{argument}(\mathtt{n1}, \{\mathtt{a} \wedge \mathtt{b}, \mathtt{a} \rightarrow \mathtt{c}\}, \mathtt{c}) \\
\mathtt{argument}(\mathtt{n2}, \{\neg\mathtt{c}\}, \neg\mathtt{c}) \\
\mathtt{posit}(\mathtt{n1}) \\
\mathtt{posit}(\mathtt{n2}) \\
\mathtt{attacks}(\mathtt{n2}, \mathtt{n1}) \\
\mathtt{graph}(\mathtt{n1}, \mathtt{n2}, (\mathtt{n2}, \mathtt{n1}))
\end{array}
$$

We use this language for literals as a meta-language for argumentation (c.f Wooldridge, McBurney + Parsons AAMAS 2005).

Specifying dialogical argumentation

Action units

For each literal α, there are four kinds of action unit

- $\oplus\alpha$ which means that the action by an agent is to add α to its next private state.
- $\ominus\alpha$ which means that the action by an agent is to delete α from its next private state.
- $\boxplus\alpha$ which means that the action by an agent is to add α to the next public state.
- $\boxminus\alpha$ which means that the action by an agent is to delete α from the next public state.

Action rule

If ϕ is a classical formula and ψ is an action formula such that Variable(ψ) $\subseteq$ Variable(ϕ), then $\phi \rightarrow \psi$ is an **action rule**.

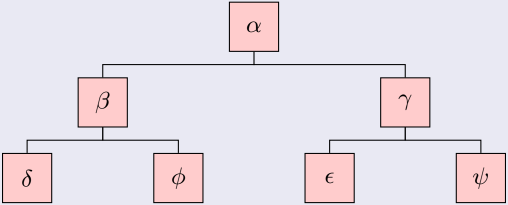
Specifying dialogical argumentation

Example of action rules with disjunction

- The action rules for each agent are the following.

  1. $\alpha \Rightarrow \boxminus\alpha \wedge (\boxplus\beta \vee \boxplus\gamma)$
  2. $\beta \Rightarrow \boxminus\beta \wedge (\boxplus\delta \vee \boxplus\phi)$
  3. $\gamma \Rightarrow \boxminus\beta \wedge (\boxplus\epsilon \vee \boxplus\psi)$

- Each branch refers to an execution (i.e. a dialogue).
- Each node in this tree is a public state.



Specifying dialogical argumentation

Example of execution updating private states

- The action rules are the following.

  Agent 1 $\alpha \wedge \delta \Rightarrow \boxplus\alpha \wedge \ominus\delta$
  Agent 2 $\alpha \Rightarrow \oplus\alpha \wedge \boxminus\beta \wedge \ominus\beta$

- There is a single execution if the starting state is

$$(\{\alpha, \delta\}, \{\}, \{\beta\}, \{\}, \{\beta\})$$

n	$s_1(n)$	$a_1(n)$	$p(n)$	$a_2(n)$	$s_2(n)$
0	α, δ		β		β
1	α, δ	$\boxplus\alpha, \ominus\delta$	β		β
2	α		α, β	$\oplus\alpha, \boxminus\beta, \ominus\beta$	β
3	α		α		α

Specifying dialogical argumentation

Example of argumentation with no concession

n	$s_1(n)$	$a_1(n)$	$p(n)$	$a_2(n)$	$s_2(n)$
0	$\mathtt{bel(p)}$		$\mathtt{turn(ann)}$		$\mathtt{bel(\neg p)}$
1	$\mathtt{bel(p)}$	$\boxplus\mathtt{claim(ann, p)}$ $\boxminus\mathtt{turn(ann)}$ $\boxplus\mathtt{turn(bob)}$	$\mathtt{turn(ann)}$		$\mathtt{bel(\neg p)}$
2	$\mathtt{bel(p)}$		$\mathtt{claim(ann, p)}$ $\mathtt{turn(bob)}$	$\boxplus\mathtt{claim(bob, \neg p)}$ $\boxminus\mathtt{turn(bob)}$ $\boxplus\mathtt{turn(ann)}$	$\mathtt{bel(\neg p)}$
3	$\mathtt{bel(p)}$	$\boxplus\mathtt{claim(ann, p)}$ $\boxminus\mathtt{turn(ann)}$ $\boxplus\mathtt{turn(bob)}$	$\mathtt{claim(ann, p)}$ $\mathtt{claim(bob, \neg p)}$ $\mathtt{turn(ann)}$		$\mathtt{bel(\neg p)}$
4	$\mathtt{bel(p)}$		$\mathtt{claim(ann, p)}$ $\mathtt{claim(bob, \neg p)}$ $\mathtt{turn(bob)}$	$\boxplus\mathtt{claim(bob, \neg p)}$ $\boxminus\mathtt{turn(bob)}$ $\boxplus\mathtt{turn(ann)}$	$\mathtt{bel(\neg p)}$
5	...	...	...	...	...

- $\mathtt{turn(ann)} \wedge \mathtt{bel(X)} \Rightarrow \boxplus\mathtt{claim(ann, X)} \wedge \boxminus\mathtt{turn(ann)} \wedge \boxplus\mathtt{turn(bob)}$
- $\mathtt{turn(bob)} \wedge \mathtt{bel(X)} \Rightarrow \boxplus\mathtt{claim(bob, X)} \wedge \boxminus\mathtt{turn(bob)} \wedge \boxplus\mathtt{turn(ann)}$

Specifying dialogical argumentation

Motivation for selection function

In a given state of the execution, a number of actions rules can fire, and so it may be desirable select a subset to act upon.

Examples of selection functions

- **Exhaustive** Select all fired action rules.
- **Turn taking** Select all fired action rules for agent 1 on odd steps, and all fired action rules for agent 2 on even steps.
- **Non-deterministic** Select randomly a fired action rule.
- **Ranked** Select a fired action rule with highest priority

Specifying dialogical argumentation

Example of argumentation with no concession

n	$s_1(n)$	$a_1(n)$	$p(n)$	$a_2(n)$	$s_2(n)$
0	$\mathtt{bel(p)}$				$\mathtt{bel(\neg p)}$
1	$\mathtt{bel(p)}$	$\boxplus\mathtt{claim(ann, p)}$			$\mathtt{bel(\neg p)}$
2	$\mathtt{bel(p)}$		$\mathtt{claim(ann, p)}$	$\boxplus\mathtt{claim(bob, \neg p)}$	$\mathtt{bel(\neg p)}$
3	$\mathtt{bel(p)}$	$\boxplus\mathtt{claim(ann, p)}$	$\mathtt{claim(ann, p)}$ $\mathtt{claim(bob, \neg p)}$		$\mathtt{bel(\neg p)}$
4	$\mathtt{bel(p)}$		$\mathtt{claim(ann, p)}$ $\mathtt{claim(bob, \neg p)}$	$\boxplus\mathtt{claim(bob, \neg p)}$	$\mathtt{bel(\neg p)}$
5	...	...	...	...	...

- Agent 1: $\mathtt{bel(X)} \Rightarrow \boxplus\mathtt{claim(ann, X)}$
- Agent 2: $\mathtt{bel(X)} \Rightarrow \boxplus\mathtt{claim(bob, X)}$

Specifying dialogical argumentation

Example of argumentation with concession

n	$s_1(n)$	$a_1(n)$	$p(n)$	$a_2(n)$	$s_2(n)$
0 0	$\texttt{bel}(\texttt{p})$ $\texttt{inc}(\texttt{p}, \neg\texttt{p})$				$\texttt{bel}(\neg\texttt{p})$
1	$\texttt{bel}(\texttt{p})$ $\texttt{inc}(\texttt{p}, \neg\texttt{p})$	$\boxplus\texttt{claim}(\texttt{ann}, \texttt{p})$			$\texttt{bel}(\neg\texttt{p})$
2	$\texttt{bel}(\texttt{p})$ $\texttt{inc}(\texttt{p}, \neg\texttt{p})$		$\texttt{claim}(\texttt{ann}, \texttt{p})$	$\boxplus\texttt{claim}(\texttt{bob}, \neg\texttt{p})$	$\texttt{bel}(\neg\texttt{p})$
3	$\texttt{bel}(\texttt{p})$ $\texttt{inc}(\texttt{p}, \neg\texttt{p})$	$\boxminus\texttt{claim}(\texttt{ann}, \texttt{p})$	$\texttt{claim}(\texttt{ann}, \texttt{p})$ $\texttt{claim}(\texttt{bob}, \neg\texttt{p})$		$\texttt{bel}(\neg\texttt{p})$
4	$\texttt{bel}(\texttt{p})$ $\texttt{inc}(\texttt{p}, \neg\texttt{p})$		$\texttt{claim}(\texttt{bob}, \neg\texttt{p})$		$\texttt{bel}(\neg\texttt{p})$

- Agent 1: $\texttt{bel}(\texttt{X}) \Rightarrow \boxplus\texttt{claim}(\texttt{ann}, \texttt{X})$
- Agent 1: $\texttt{claim}(\texttt{ann}, \texttt{X}) \wedge \texttt{claim}(\texttt{bob}, \texttt{Y}) \wedge \texttt{inc}(\texttt{X}, \texttt{Y}) \Rightarrow \boxminus\texttt{claim}(\texttt{ann}, \texttt{X})$
- Agent 2: $\texttt{bel}(\texttt{X}) \Rightarrow \boxplus\texttt{claim}(\texttt{bob}, \texttt{X})$

Specifying dialogical argumentation

Using builtin predicates in action rules

For the condition of an action rule, we want to include literals that are "builtin" predicates (as in logic programming)

$$\texttt{query}(\texttt{X}) \wedge \texttt{argument}(\texttt{S},\texttt{X}) \Rightarrow \boxplus\texttt{posit}(\texttt{S},\texttt{X})$$

Examples of definitions for builtin predicates

$\texttt{argument}(\texttt{S},\texttt{C}) :- \ \texttt{beliefs}(\texttt{B}), \texttt{subset}(\texttt{S},\texttt{B}), \texttt{entails}(\texttt{S},\texttt{C}).$

$\texttt{beliefs}(\texttt{B}) :- \ \texttt{findall}(\texttt{Y}, \texttt{belief}(\texttt{Y}), \texttt{B}).$

$\texttt{entails}(\texttt{S},\texttt{C}) :- \ \texttt{literal}(\texttt{C}), \texttt{member}(\texttt{C},\texttt{S}).$
$\texttt{entails}(\texttt{S},\texttt{C}) :- \ \texttt{member}(\texttt{R},\texttt{S}), \texttt{rule}(\texttt{R}), \texttt{head}(\texttt{R},\texttt{C}), \texttt{tail}(\texttt{R},\texttt{T}), \texttt{entails}(\texttt{S},\texttt{T}).$

$\texttt{literal}(\texttt{X}) :- \ \texttt{atom}(\texttt{X}).$
$\texttt{literal}(\neg\texttt{X}) :- \ \texttt{atom}(\texttt{X}).$

Specifying dialogical argumentation

Specification for a dialogical argumentation system in executable logic

- A set of definitions for builtin predicates.
- A set of action rules for each agent.
- A selection function for each agent.
- A set of initial states.

[See Black and Hunter ECAI 2012]

Specifying dialogical argumentation

Assume the following variables and function.

- $\mathtt{X}$ is an agent and $\mathtt{op(X)}$ is the other agent.
- $\mathtt{B}$ and $\mathtt{C}$ are formulae, and $\mathtt{S}$ is a set of formulae.

Action rules

1. $\mathtt{claim(X,C)} \wedge \mathtt{unqueried(C)}$
   $\Rightarrow \boxplus\mathtt{why(op(X),C)}$
2. $\mathtt{claim(X,C)} \wedge \mathtt{why(op(X),C)} \wedge \mathtt{lacksarg(X,C)}$
   $\Rightarrow \boxminus\mathtt{claim(X,C)} \wedge \boxminus\mathtt{why(op(X),C)}$
3. $\mathtt{why(op(X),C)} \wedge \mathtt{hasarg(X,S,C)} \wedge \mathtt{new(S,C)}$
   $\Rightarrow \boxplus\mathtt{explain(X,S,C)} \wedge \boxminus\mathtt{why(op(X),C)}$
4. $\mathtt{explain(X,S,C)} \wedge \mathtt{in(B,S)} \wedge \mathtt{bel(B)} \wedge \mathtt{unconceded(op(X),B)}$
   $\Rightarrow \boxplus\mathtt{concede(op(X),B)}$
5. $\mathtt{explain(X,S,C)} \wedge (\mathtt{in(B,S)} \vee \mathtt{B} = \mathtt{C}) \wedge \mathtt{hasarg(op(X),T,\neg B)} \wedge \mathtt{new(T,\neg B)}$
   $\Rightarrow \boxplus\mathtt{explain(op(X),T,\neg B)}$

Specifying dialogical argumentation

Example of a persuasion dialogue

n	x	$a_x(n)$
0	1	$\boxplus\mathtt{claim}(\mathtt{Paul}, \mathtt{safe})$
1	2	$\boxplus\mathtt{why}(\mathtt{Olga}, \mathtt{safe})$
2	1	$\boxminus\mathtt{why}(\mathtt{Olga}, \mathtt{safe})$ $\boxplus\mathtt{explain}(\mathtt{Paul}, \{\mathtt{airbag}, \mathtt{airbag} \rightarrow \mathtt{safe}\}, \mathtt{safe})$
3	2	$\boxplus\mathtt{concede}(\mathtt{Olga}, \mathtt{airbag})$ $\boxplus\mathtt{explain}(\mathtt{Olga}, \{\mathtt{explosions}, \mathtt{explosions} \rightarrow \neg\mathtt{safe}\}, \neg\mathtt{safe})$ $\boxplus\mathtt{explain}(\mathtt{Olga}, \{\mathtt{highspeed}, \mathtt{highspeed} \rightarrow \neg\mathtt{safe}\}, \neg\mathtt{safe})$
4	1	$\boxplus\mathtt{explain}(\mathtt{Paul}, \{\mathtt{unreliablenews}, \mathtt{unreliablenews} \rightarrow \neg\mathtt{explosions}\}, \neg\mathtt{explosions})$

$$\Delta_1 = \{\mathtt{airbag}, \mathtt{airbag} \rightarrow \mathtt{safe}, \mathtt{unreliablenews}, \mathtt{unreliablenews} \rightarrow \neg\mathtt{explosions}\}$$

$$\Delta_2 = \{\mathtt{explosions}, \mathtt{explosions} \rightarrow \neg\mathtt{safe}, \mathtt{highspeed}, \mathtt{highspeed} \rightarrow \neg\mathtt{safe}\}$$

Specifying dialogical argumentation

Example of a collaborative dialogue

n	x	$a_x(n)$
0	1	$\boxplus\mathtt{query}(\mathtt{John}, \mathtt{p})$
1	2	$\boxplus\mathtt{suggest}(\mathtt{Mary}, \mathtt{q} \rightarrow \mathtt{p})$
2	1	$\boxplus\mathtt{suggest}(\mathtt{John}, \mathtt{r} \rightarrow \mathtt{q})$
3	2	$\boxplus\mathtt{skip}(\mathtt{Mary})$
4	1	$\boxplus\mathtt{suggest}(\mathtt{John}, \mathtt{s} \rightarrow \mathtt{q})$
5	2	$\boxplus\mathtt{suggest}(\mathtt{Mary}, \mathtt{s})$ $\boxminus\mathtt{skip}(\mathtt{Mary})$
6	1	$\boxplus\mathtt{posit}(\mathtt{John}, \langle\{\mathtt{s}, \mathtt{s} \rightarrow \mathtt{q}, \mathtt{q} \rightarrow \mathtt{p}\}, \mathtt{p})$ $\boxminus\mathtt{query}(\mathtt{John}, \mathtt{p})$

- $\Delta_1 = \{\mathtt{r} \rightarrow \mathtt{q}, \mathtt{s} \rightarrow \mathtt{q}\}$
- $\Delta_2 = \{\mathtt{q} \rightarrow \mathtt{p}, \mathtt{s}\}$

Specifying dialogical argumentation

Theoretical analysis of specifications

Given a specification for a dialogical argumentation system, it is relatively straightforward to prove properties concerning

- Termination
- Possible/guaranteed success (w.r.t. goals)
- Consistent states (actions,beliefs)
- Comparison of rules (e.g. redundancy, equivalence, etc)
- Equivalence of systems
- Dynamics (e.g. deadlock, fairness, etc).
- Satisfaction of protocols

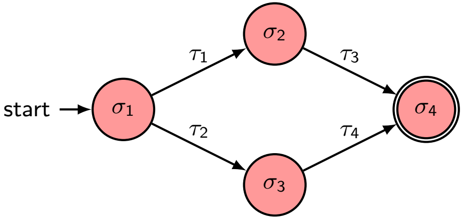
Empirical analysis of specifications

Action rules are executable via an execution engine that can handle the builtin predicates (e.g. code for builtin predicates, code to subcontract evaluation of builtin predicates to Prolog, etc.)

Specifying dialogical argumentation

$$\alpha \Rightarrow (\boxplus\beta \vee \boxplus\gamma) \wedge \ominus\alpha$$
$$\beta \Rightarrow \boxplus\delta \wedge \boxminus\beta$$
$$\gamma \Rightarrow \boxplus\delta \wedge \boxminus\gamma$$

The FSM below fabricates the action rules above with $(\{\alpha\}, \{\}, \{\}, \{\}, \{\})$ as the starting state.



σ_1 is $(\{\alpha\}, \{\}, \{\})$
σ_2 is $(\{\}, \{\beta\}, \{\})$
σ_3 is $(\{\}, \{\gamma\}, \{\})$
σ_4 is $(\{\}, \{\delta\}, \{\})$

τ_1 is $(\{\boxplus\beta, \ominus\alpha\}, \{\})$
τ_2 is $(\{\boxplus\gamma, \ominus\alpha\}, \{\})$
τ_3 is $(\{\boxminus\beta, \boxplus\delta\}, \{\boxminus\beta, \boxplus\delta\})$
τ_4 is $(\{\boxminus\gamma, \boxplus\delta\}, \{\boxminus\gamma, \boxplus\delta\})$

Specifying dialogical argumentation

Properties of propositional action rules

- For each finite state machine M, there is a set of action rules that simulate M.
- For each set of propositional action rules S, and each starting state c, there is a finite state machine M such that M fabricates S w.r.t. c.

Observations

1. Most proposals for formalizing dialogical argumentation are essentially propositional, therefore reducing them to finite state machines offers computational viability.
2. Yet the potential for using first-order builtin predicates for more sophisticated behaviours suggests we will go well beyond propositional action rules, and therefore there is some exciting research to do (e.g. strategies, taking the audience into account, persuasion, etc.).

Strategies

Current situation

Most research to date has focused on simple strategies such as being exhaustive in presenting arguments and/or being co-operative in presenting knowledge that may be useful to the other agents

New research directions

Now interest is turning to more sophisticated strategies which allow an agent to choose which moves to make in order to optimize its outcome from the dialogue.

- Being economical with the truth.
- Selecting moves that the audience is more likely to believe, or value, or that address its goals.
- Selecting sequences of moves that are more likely to be persuasive.

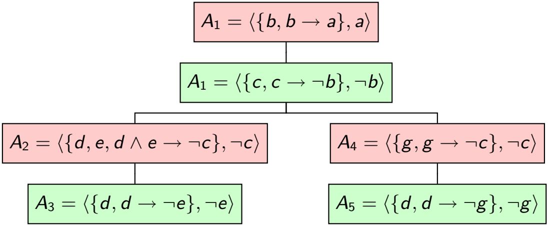
[See for example Black & Hunter JAAMAS 2009; Fan & Toni ECAI 2012; Fan & Toni COMMA 2012; etc.]

Strategies

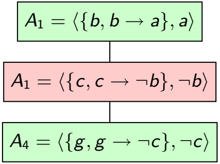
Consider the following knowledgebases for each agent

- $\Delta_1 = \{b, d, e, g, b \rightarrow a, d \wedge e \rightarrow \neg c, g \rightarrow \neg c\}$
- $\Delta_2 = \{c, c \rightarrow \neg b, d \rightarrow \neg e, d \rightarrow \neg g\}$

The left argument graph is obtained when agent 1 uses a naive strategy and the right argument graph is obtained when agent 1 uses a more intelligent strategy.



Agent 1 is exhaustive in the arguments posited, thereby allowing agent 2 to construct arguments that cause the root to be defeated.



Agent is selective in the arguments posited, thereby ensuring that the root as undefeated.

Modelling audiences

Example of analysing empathy and antipathy

- t = "Need to raise federal taxes"
- r = "Need to invest in road and rail infrastructure"
- h = "Need to build more hospitals"

$$\langle\{r, r \rightarrow t\}, t\rangle \qquad \langle\{h, h \rightarrow t\}, t\rangle$$

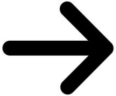
If the audience believes $\{r, \neg h\}$, then they will have an empathy for the left argument and antipathy for the right argument.

Example involving ethical values

Consider the following arguments concerning the theft by Hal of insulin from Carla's House because he has lost his through no fault of his own.

- A_1 = "Hal is justified because a person can use other people's property to save a life" (LIFE)
- A_2 = "It is wrong to infringe the property rights of others" (PROPERTY)

Towards systems for intelligent interactions



- Arguments and counterarguments are important feature of intelligent interactions.
- Computational models of argument can be part of the solution for developing intelligent interactions
    - In the short term, argumentation through structured interactions (e.g. clinical decision support systems, web systems for consultations in eGovernment, etc)
    - In the longer term, argumentation will need to be integrated with natural language processing, intelligent user models, etc, to develop systems with truely intelligent interactions.

Towards applications in autonomous systems

- Heterogeneous robots need to work together to survey a situation.
- Exchanging low level information wastes time and bandwidth, and may not be possible if not designed to do so.
- Exchanging high level information (e.g. knowledge-level sensor fusion) means that the robots will need to deal with incomplete and inconsistent information.
- Working together via argumentation means high level information can be exchanged.
- So argumentation offers basis for "USB technology" for intelligent robots.

Conclusions

- Argumentation is an important cognitive process for dealing with incomplete and inconsistent information.
- There is a range of formal models of dialogical argumentation which investigate protocols and strategies.
- Most current proposals for dialogical argumentation are "essentially propositional" and use "naive strategies" for each agent.
- Key research questions in developing computational models of dialogical argumentation include
    - Strategies that improve the outcome of the dialogue for the agent's benefit
    - Better modelling of the recipents of dialogue moves (audience modelling)
    - Better modelling of persuasion