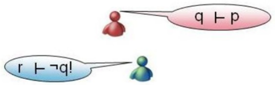
From logic programming and monotonic reasoning to computational argumentation and beyond

Francesca Toni

Department of Computing, Imperial College London, UK
Computational Logic and Argumentation Group



LPNMR 2017, 4 July 2017

Computational Argumentation (aka Argumentation in AI)

Non-Monotonic Reasoning (NMR) and Logic Programming (LP)

from late 1980s (e.g. Lin, Shoham, Dung, Kowalski, Kakas, Toni):
$\Rightarrow$ abstract argumentation, ABA

Defeasible Reasoning as studied in philosophy

from late 1980s (e.g. Pollock, Nute):
$\Rightarrow$ DeLP, ASPIC, ASPIC+

Decision making

from early 1990s (e.g. Fox, Krause, Ambler):
$\Rightarrow$ Amgoud and Prade (2009), ...

Resolving inconsistencies (paraconsistent reasoning)

from mid 1990s (e.g. Cayrol, Amgoud, Hunter):
$\Rightarrow$ logic-based argumentation

Outline

- LP with negation as failure (NAF) – and several NMR formalisms – can be understood in terms of:
  - abstract argumentation (AA) [Dung 95]
  - assumption-based argumentation (ABA) [Bondarenko et al 97]
- computational argumentation tools generalise/use LP tools:
  - (top-down) dispute trees/derivations [Dung et al 06,07, Toni 13] vs SLD-based computation in LP
  - (bottom-up) computation of AA extensions via ASP [Toni, Sergot 11 - survey]
- computational argumentation for
  - explanations, e.g. of (non-)membership in answer sets, decisions, outcomes of case-based reasoning
  - collaborative decision-making

LP for NMR – "uncontroversial" examples

Example

$p \leftarrow not\ q$

all LP semantics agree:
p holds (because) q doesn't

Example

$p \leftarrow not\ q$
$q \leftarrow not\ r$

all LP semantics agree:
p doesn't hold (because) q does (because) r doesn't

LP for NMR: "controversial" examples

Example (two-loop: $p \leftarrow not\ q$, $q \leftarrow not\ p$)

two answer sets: either p or q
"empty" well-founded model: neither p nor q

Example (one-loop: $r \leftarrow not\ r$, $p \leftarrow not\ q$)

no answer set
well-founded model: p
one partial stable model: p

Example (two-loop+one-loop: $p \leftarrow not\ q$, $q \leftarrow not\ p$, $r \leftarrow not\ r$)

no answer set
"empty" well-founded model
two partial stable models: either p or q, neither r nor $not\ r$

Other NMR Formalisms

Presumption of innocence: *a person is innocent unless proven guilty*. Mary is a person (accused of some crime): should Mary be deemed innocent? **yes, no matter which formalism** e.g.[1]

LP: $i(mary)$ holds given:

$i(X) \leftarrow p(X), not\ g(X)$ $\qquad$ $p(mary) \leftarrow$

Default Logic: $i(mary)$ holds given:

$D : \dfrac{p(mary) : M\neg g(mary)}{i(mary)}$ $\qquad$ $W : p(mary)$

Non-Monotonic Modal Logic: $i(mary)$ holds given:

$p(mary) \wedge \neg L g(mary) \rightarrow i(mary)$ $\qquad$ $p(mary)$

[1] i=innocent, g=guilty, p=person

Argumentation for LP and NMR: intuition

LP: $i(X) \leftarrow p(X), not\ g(X), \quad p(mary) \leftarrow$

- there is an **argument** for $i(mary)$ supported by $not\ g(mary)$
- there is no objection (**attack**) against this argument
- the argument is thus "**acceptable**"

Default Logic: $D : \dfrac{p(mary) : M\neg g(mary)}{i(mary)}, \ W : p(mary)$

- there is an **argument** for $i(mary)$ supported by $M\neg g(mary)$
- there is no objection (**attack**) against this argument
- the argument is thus "**acceptable**"

NM Modal Logic: $p(mary) \wedge \neg Lg(mary) \rightarrow i(mary), \quad p(mary)$

- there is an **argument** for $i(mary)$ supported by $\neg Lg(mary)$
- ...

Abstract Argumentation (AA)

An AA framework is a pair $\langle Args, attacks \rangle$ where

- $Args$ is a set (the *arguments*)
- $attacks \subseteq Args \times Args$ is a binary relation over $Args$

$(\alpha, \beta) \in attacks$ is written/read as "α attacks β"

An AA framework can be represented as a directed graph

Example (*attacks* represented by directed edges)

$$i \longleftarrow g_{as} reliable.witness \longleftarrow \neg reliable.witness_{as} drunk$$

AA for LP

Given a program P, let $\langle Args, attacks \rangle$ be the AA framework where

- *arguments* (in *Args*) are deductions from P (Modus Ponens with $\leftarrow$) and NAF literals (treated as abducibles)
- $P \cup \Delta \vdash x$ *attacks* $P \cup \Gamma \vdash y$ iff $not\ x \in \Gamma$

Example ($P:\ p \leftarrow not\ q, q \leftarrow not\ r$)

Arguments include $P \cup \{not\ q\} \vdash p$,
$P \cup \{not\ r\} \vdash q$,
$P \cup \{not\ p\} \vdash not\ p$

attacks includes $P \cup \{not\ r\} \vdash q$ *attacks* $P \cup \{not\ q\} \vdash p$
$P \cup \{not\ q\} \vdash p$ *attacks* $P \cup \{not\ p\} \vdash not\ p$

$$\text{arg}(not\ p) \longleftarrow \text{arg}(p) \longleftarrow \text{arg}(q)$$

Semantics for AA

Recipes for "acceptable" sets of arguments (extensions)[2]

$A \subseteq$ *Args* is

- *conflict-free* iff it does not attack itself
- *stable* iff it is conflict-free and attacks every $\alpha \in Args \setminus A$
- *admissible* iff it is conflict-free and attacks back each attacking argument; *preferred* iff it is maximal (wrt $\subseteq$) admissible
- *complete* iff it is admissible and contains all arguments it defends (by attacking all attacks against them); *grounded* iff it is minimal (wrt $\subseteq$) complete

Example ($\langle Args, attacks \rangle$ is $\gamma \longrightarrow \beta \longrightarrow \alpha$)

$\{\alpha, \gamma\}$ is stable, preferred, complete, grounded

[2] $A \subseteq Args$ attacks $B \subseteq Args$ iff $\alpha \in A$ attacks $\beta \in B$;
$A \subseteq Args$ attacks $\beta \in Args$ iff A attacks $\{\beta\}$

Semantics for AA

Recipes for "acceptable" sets of arguments (extensions)[3]

$A \subseteq Args$ is

- *conflict-free* iff it does not attack itself
- *stable* iff it is conflict-free and attacks every $\alpha \in Args \setminus A$
- *admissible* iff it is conflict-free and attacks back each attacking argument; *preferred* iff it is maximal (wrt $\subseteq$) admissible
- *complete* iff it is admissible and contains all arguments it defends (by attacking all attacks against them); *grounded* iff it is minimal (wrt $\subseteq$) complete

Example ($\langle Args, attacks \rangle$ is $\alpha \leftrightarrow \beta$)

$\{\alpha\}$ is stable, preferred, complete (and so is $\{\beta\}$)
$\{\}$ is grounded

[3] $A \subseteq Args$ attacks $B \subseteq Args$ iff $\alpha \in A$ attacks $\beta \in B$; $A \subseteq Args$ attacks $\beta \in Args$ iff A attacks $\{\beta\}$

AA Semantics vs LP Semantics

Given P, let AA_P be the AA framework correponding to P:

- stable extensions of AA_P correspond to **answer sets** of P:
  - S is a stable extension of AA_P iff $\{x|P \cup \_ \vdash x \in S\}$ is an answer set of P
  - given interpretation M of P, let $\Delta_M = \{not\ x|x \notin M\}$:
    M is an answer set of P iff $\{P \cup \Delta \vdash x|\Delta \subseteq \Delta_M\}$ is a stable extension of AA_P
- preferred extensions of AA_P correpond to (Saccà and Zaniolo's) **partial stable models** of P
- complete extensions of AA_P correspond to (Przymusinski's) **3-value stable models** of P
- the grounded extension corresponds to the **well-founded model** of P

AA for LP: Some Examples

Example ($P : p \leftarrow not\ q, \quad q \leftarrow not\ p$)

- $\{P \cup \Delta \vdash x | not\ q \in \Delta\}$ is stable/preferred (e.g. $\Delta = \{not\ q\}$, $x = not\ q$)
- $\{P \cup \Delta \vdash x | not\ p \in \Delta\}$ is stable/preferred (e.g. $\Delta = \{not\ p\}$, $x = q$)
- $\{\}$ is grounded

Example ($P : p \leftarrow not\ q, \quad r \leftarrow not\ r$)

- no stable extension ($P \cup \{not\ r\} \vdash r$ can be neither in nor out)
- one preferred/grounded extension: $\{P \cup \Delta \vdash x | not\ q \in \Delta\}$ (e.g. $\Delta = \{not\ q\}$, $x = p$)

Assumption-Based Argumentation (ABA)

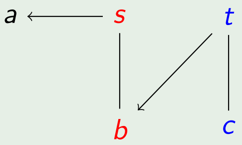
An *ABA framework* is a tuple $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ where

- $\langle \mathcal{L}, \mathcal{R} \rangle$ is a deductive system with language $\mathcal{L}$ and rules $\mathcal{R}$
- $\mathcal{A} \subseteq \mathcal{L}$ are *assumptions*
- $\overline{}$ is a total mapping from $\mathcal{A}$ into $\mathcal{L}$, $\overline{\alpha}$ is the *contrary* of α

Arguments are trees - deductions (wrt $\langle \mathcal{L}, \mathcal{R} \rangle$) of claims supported by sets of assumptions.
Attacks are directed at the assumptions in the support of arguments – by deriving their contrary.

Example ($\mathcal{R} = \{s \leftarrow b, t \leftarrow c\}$, $\mathcal{A} = \{a, b, c\}$, $\overline{a} = s, \overline{b} = t, \overline{c} = u$)

arguments include $\{a\} \vdash a$
$\{b\} \vdash s$
$\{c\} \vdash t$



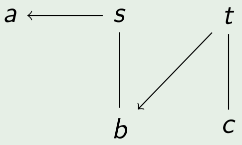
ABA semantics

Flat ABA ($\approx$ no assumption is the head of a rule):

- stable, preferred, grounded etc sets of arguments – as in AA
- stable, preferred, grounded etc sets of assumptions

The two views (argument view and assumption view) correspond

Example ($\mathcal{R} = \{s \leftarrow b, t \leftarrow c\}$, $\mathcal{A} = \{a, b, c\}$, $\overline{a} = s, \overline{b} = t, \overline{c} = u$)



$\{b\}$ attacks $\{a\}$
$\{c\}$ attacks $\{b\}$

- $\{c, a\}$ is a stable etc set of assumptions
- the set of all arguments supported by subsets of $\{c, a\}$ is a stable etc extension

Note: flat ABA is an instance of AA; AA is an instance of flat ABA

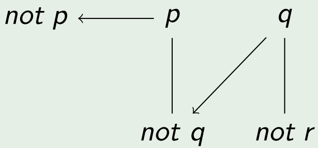
ABA for LP

Given P, ABA_P is the ABA framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ where

- $\mathcal{R}$ is P
- $\mathcal{L} = HB_P \cup HB_P^{NAF}$ where HB_P is the Herbrand Base of P and HB_P^{NAF} is the set of all NAF literals over HB_P
- $\mathcal{A} = HB_P^{NAF}$, $\overline{not\ x} = x$

Example ($P:\ p \leftarrow not\ q, q \leftarrow not\ r$)

- $\mathcal{R} = \{p \leftarrow not\ q, q \leftarrow not\ r\}$
- $\mathcal{L} = \{p, q, r, not\ p,\ not\ q,\ not\ r\}$
- $\mathcal{A} = \{not\ p,\ not\ q,\ not\ r\}$, $\overline{not\ p} = p$, $\overline{not\ q} = q$, $\overline{not\ r} = r$



AA/ABA for other NMR formalisms

- Different kinds of arguments and attacks
- same semantics (stable extensions)

Example (Default Logic, presumption of innocence)

$$D: \quad \frac{p(mary) : M\neg g(mary)}{i(mary)}, \quad \frac{witness\_against(john, mary) : Mreliable(john)}{g(mary)}$$

$$W: \quad p(mary),\ witness\_against(john, mary)$$

- AA: $D \cup W \cup \{Mreliable(john)\} \vdash_{FOL+D} g(mary)$ attacks $D \cup W \cup \{M\neg g(mary)\} \vdash_{FOL+D} i(mary)$
- ABA: $\{Mreliable(john)\}$ attacks $\{M\neg g(mary)\}$

Computational argumentation tools

Based on

- dispute trees and dispute derivations (vs SLD-based computation in LP: `proxdd`, `grapharg`)
- mapping of computation of extensions onto (meta-)logic programs (vs ASP: `ASPARTIX`)
- (constraint solving: `Conarg`)

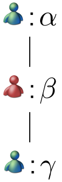
Dispute trees

Given $\langle Args, attacks \rangle$, a *dispute tree* for $\alpha \in Args$ is a tree $\mathcal{T}$ s.t.

1. each node of $\mathcal{T}$ is labelled by some $\chi \in Args$ and is by the *proponent* or the *opponent*
2. the root of $\mathcal{T}$ is a node labelled by α;
3. for each node *n*, labelled by some $\beta \in Args$, and for every $(\gamma, \beta) \in attacks$ there is a child of *n* labelled by γ
4. for each node *n*, labelled by some $\beta \in Args$, there is *at most* one child of *n* which is by and labelled by some γ s.t. $(\gamma, \beta) \in attacks$
5. there are no other nodes in $\mathcal{T}$

The *defence set* of $\mathcal{T}$ is the set of all its arguments

Given $\gamma \rightarrow \beta \rightarrow \alpha$



Different types of dispute trees

A dispute tree is

- *admissible* iff (i) every  node has exactly one child, and
  (ii) no argument labels both  and  nodes.
- *grounded* iff (i) every  node has exactly one child, and
  (ii) it is finite

Theorem

The defence set of an admissible/grounded dispute tree is admissible/contained in the grounded extension

Dispute derivations to compute (different types of) dispute trees

Example of X-dispute derivation (X=admissible/grounded)

Given $\mathcal{R} = \{p \leftarrow not\ q,\ q \leftarrow not\ r\}$, $\mathcal{A} = \{not\ p, not\ q, not\ r\}$, $\overline{not\ x} = x$:

			Defences	Culprits
0	$\{\{\} \vdash_{\{not\ p\}} not\ p\}$	$\{\}$	$\{not\ p\}$	$\{\}$
1	$\{\}$	$\{\{\} \vdash_{\{p\}} p\}$	$\{not\ p\}$	$\{\}$
2	$\{\}$	$\{\{\} \vdash_{\{not\ q\}} p\}$	$\{not\ p\}$	$\{\}$
3	$\{\{\} \vdash_{\{q\}} q\}$	$\{\}$	$\{not\ p\}$	$\{not\ q\}$
4	$\{\{\} \vdash_{\{not\ r\}} q\}$	$\{\}$	$\{not\ p, not\ r\}$	$\{not\ q\}$
5	$\{\}$	$\{\{\} \vdash_{\{r\}} r\}$	$\{not\ p, not\ r\}$	$\{not\ q\}$
6	$\{\}$	$\{\}$	$\{not\ p, not\ r\}$	$\{not\ q\}$

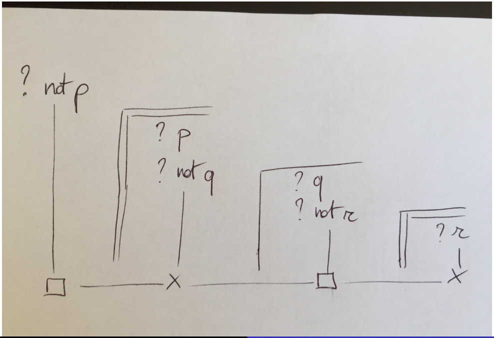
Left	Step	Right
$: \{not\ p\} \vdash_{\{\}} not\ p$	$[Step\ 1]$	$: \{not\ p\} \vdash d$
$: \{not\ q\} \vdash_{\{\}} p$	$[Step\ 3]$	$: \{not\ q\} \vdash p$
$: \{not\ r\} \vdash_{\{\}} q$	$[Step\ 5]$	$: \{not\ r\} \vdash q$

Left: the computed *dialectical tree* (of *potential* arguments)

Right: the computed *dispute tree* (of *actual* arguments)

X-dispute derivation (X=admissible/grounded) vs SLDNF/EK abductive proof procedure

			Defences	Culprits
0	$\{\{\} \vdash_{\{not\ p\}} not\ p\}$	$\{\}$	$\{not\ p\}$	$\{\}$
1	$\{\}$	$\{\{\} \vdash_{\{p\}} p\}$	$\{not\ p\}$	$\{\}$
2	$\{\}$	$\{\{\} \vdash_{\{not\ q\}} p\}$	$\{not\ p\}$	$\{\}$
3	$\{\{\} \vdash_{\{q\}} q\}$	$\{\}$	$\{not\ p\}$	$\{not\ q\}$
4	$\{\{\} \vdash_{\{not\ r\}} q\}$	$\{\}$	$\{not\ p, not\ r\}$	$\{not\ q\}$
5	$\{\}$	$\{\{\} \vdash_{\{r\}} r\}$	$\{not\ p, not\ r\}$	$\{not\ q\}$
6	$\{\}$	$\{\}$	$\{not\ p, not\ r\}$	$\{not\ q\}$



Computation of extensions via ASP

- $\langle Args, attacks \rangle$ is mapped onto a logic program, e.g. $P_{\langle Args, attacks \rangle}$ with clauses
  - $arg(\alpha) \leftarrow$ for all $\alpha \in Args$ and
  - $att(\alpha, \beta) \leftarrow$ for all $(\alpha, \beta) \in attacks$
- semantics correspond to answer sets of logic programs, e.g. in ASPARTIX [Egly et al 08] let

$$\begin{aligned} P_{cf}: \quad & \leftarrow in(X), in(Y), att(X,Y), \\ & in(X) \leftarrow not\ out(X), arg(X), \\ & out(X) \leftarrow not\ in(X), arg(X) \end{aligned}$$

A is a *conflict-free extension* of $\langle Args, attacks \rangle$ iff $A = \{\alpha | in(\alpha) \in S\}$ for some answer set S of $P_{\langle Args, attacks \rangle} \cup P_{cf}$

Computational argumentation for . . .

- explanation
- collaborative decision-making

ABA-Based Answer Set Justification [Schulz,Toni 16]

Why is a literal in an answer set of a (consistent) logic program?
IDEA: literal l in answer set iff argument with conclusion l in stable extension $\Rightarrow$ use admissible dispute tree for argument with conclusion l to explain l

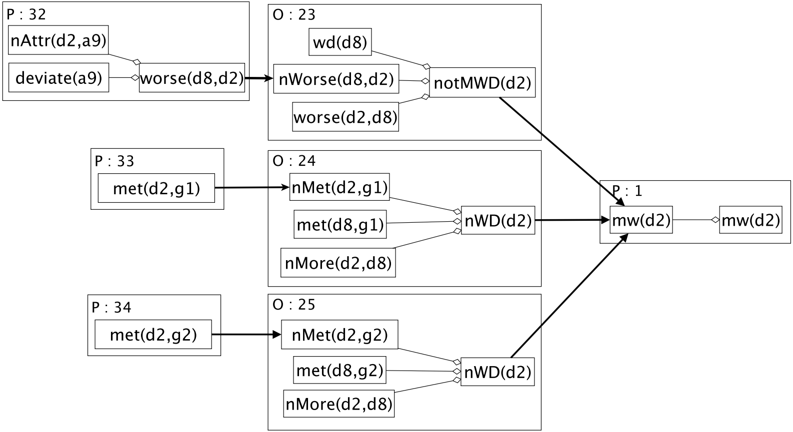
Two types of justifications:

1. argument view - **Attack Tree** (cf admissible dispute tree)
2. literal view - **LABAS Justification** (Labelled ABA-Based Answer Set Justification) - extracted from Attack Tree

Why is a literal **not** in an answer set of a (consistent) logic program? Also two types of justifications . . .

Explaining decisions [Fan et al 14, Zhong et al 14]

P= , O= :



d2 is "best" as it meets goals g1 and g2 and, moreover, d8, that also meets g1 and g2, unnecessarily has attribute a9

AA-CBR [Cyras et al 16]

Case-based Reasoning (CBR)

- Given *past cases* (S, o) (S features, $o \in \{+, -\}$ outcome)
  e.g. $(\{ensuite, wireless\}, +)$, $(\{small\}, -)$
- a default outcome $d \in \{+, -\}$ e.g. $d = +$
- Determine the outcome of new case N e.g. $N = \{ensuite, small\}$

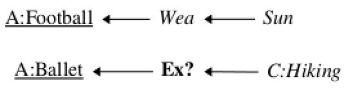
AA-CBR=CBR by mapping onto AA:

- Arguments: past cases, $(N, ?)$, $(\emptyset, d)$
  e.g. $(\{ensuite, wireless\}, +)$, $(\{small\}, -)$, $(\{ensuite, small\}, ?)$, $(\emptyset, +)$
- Attack by $\neq$outcome&specificity&coincision/irrelevance:
  e.g. $(\{small\}, -)$ attacks $(\emptyset, +)$, $(\{ensuite, small\}, ?)$ attacks $(\{ensuite, wireless\}, +)$
- outcome of N is d ($\overline{d}$) if $(\emptyset, d)$ is (not) in grounded extension
  e.g. the outcome for $N = \{ensuite, small\}$ is $-$
- dispute trees as explanations of outcomes

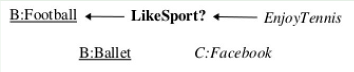
Decisions in collaborative MAS [Gao et al 16]

Information sharing, conflict resolution and privacy preservation.

Example (Variant of "Battle of the sexes")



Alice's (internal) AAF



Bob's (internal) AAF

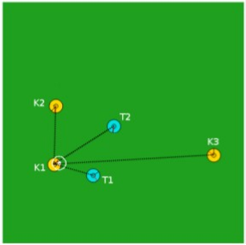
private practical, **private epistemic**, *disclosable epistemic* arguments
restrictions on attacks: practical args do not attack epistemic args, . . .
there may be attacks across, e.g. *C: Facebook* attacks *C: Hiking*

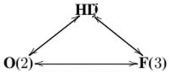
- distributed constraint satisfaction algorithm (with backtracking), incorporating variant of TPI-dispute to exchange (disclosable!) "compact reasons" drawn from explanations

A: C says she will be hiking with your ex-wife today ({*C: Hiking*,A:Ballet} is the only explanation for A:Ballet) B: . . .

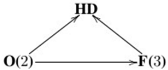
Value-based AA +Reinforcement Learning for RoboCup

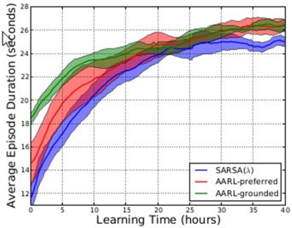
[Gao&Toni 2014]





(a) Keepaway game





(a) 2-Keepaway

Conclusions

- AA and ABA have their roots in (abductive) logic programming/non-monotonic reasoning
- dispute trees for AA/ABA can serve as the basis for explanation
- Argumentation could still "learn" from logic programming
  - non-ground engines?
- Argumentation could help LP
  - e.g. explain inconsistent logic programs under ASP?
  - e.g. explain decisions (in human-machine interactions and multi-agent systems)?