Open Research Challenges in Formal Argumentation Theory

Martin Caminada

Cardiff University
School of Computer Science

We want to express entailment as dialectics

We want to express formal argument-based entailment as the ability to win a discussion.

This is because:

- formal logic started as dialectics
- it is in line with the nature of symbolic AI
- it follows the tradition of Lorenz & Lorenzen
- foundational problem: truth and model-based semantics not suitable for NMR
- we need to go towards applications; explainability is key

Requirements

What we want is an argument game (=dialectical protocol) that:

- works with rule-based arguments
- has intuitive (user acceptable) rules regarding the allowed discussion moves
- is sound: if the proponent wins the discussion, then the main argument is accepted w.r.t. formal argumentation theory
- is complete: if the main argument is accepted w.r.t. formal argumentation theory, then the proponent has a winning strategy

Argumentation Theory: How to Construct Arguments

strict rule (→):

"from ... it always follows that..."

defeasible rule (⇒):

"from ... it usually follows that..."

$R_s = \{ \rightarrow a;\quad a \rightarrow \neg b;\quad \rightarrow d;\quad d \rightarrow e \}$

$R_d = \{ \neg b \Rightarrow c;\quad e \Rightarrow \neg c \}$

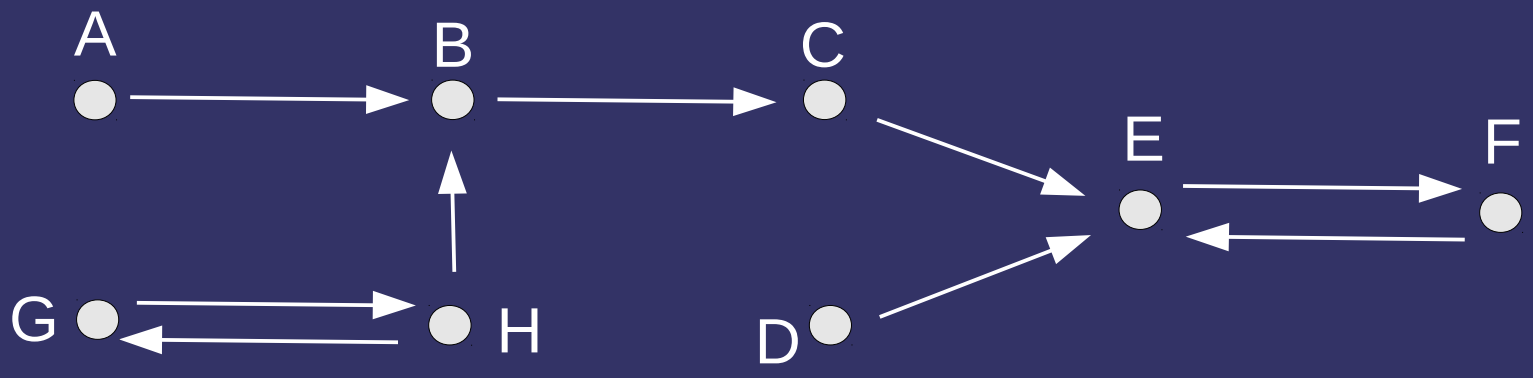
A_1: $((\rightarrow a) \rightarrow \neg b) \Rightarrow c$

A_2: $((\rightarrow d) \rightarrow e) \Rightarrow \neg c$

Argumentation Theory: How to Evaluate Arguments

complete labelling:

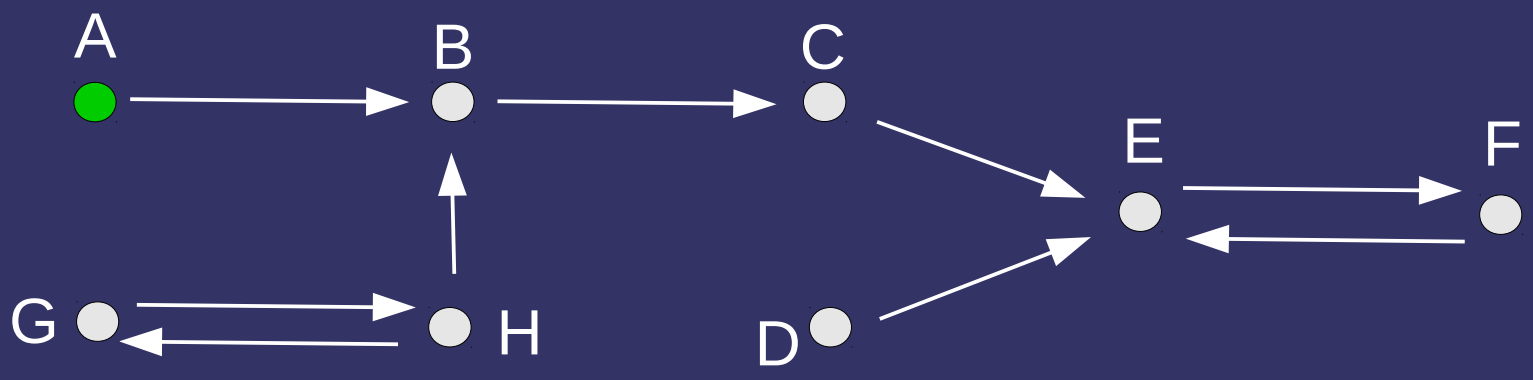
- in ⇔ all attackers out
- out ⇔ there is an attacker in
- undec ⇔ not all attackers out and no attacker in



Argumentation Theory: How to Evaluate Arguments

complete labelling:

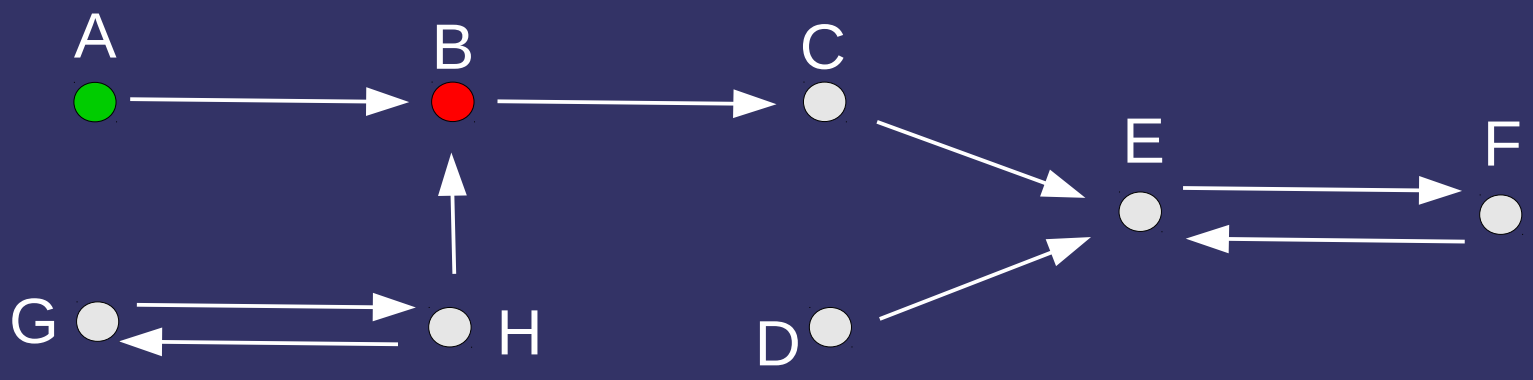
- in ⇔ all attackers out
- out ⇔ there is an attacker in
- undec ⇔ not all attackers out and no attacker in



Argumentation Theory: How to Evaluate Arguments

complete labelling:

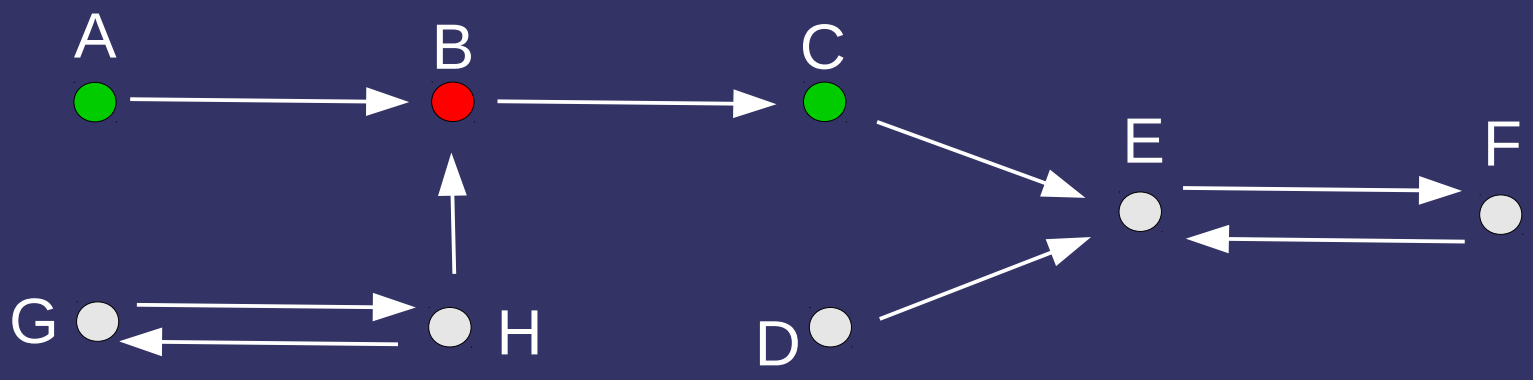
- in ⇔ all attackers out
- out ⇔ there is an attacker in
- undec ⇔ not all attackers out and no attacker in



Argumentation Theory: How to Evaluate Arguments

complete labelling:

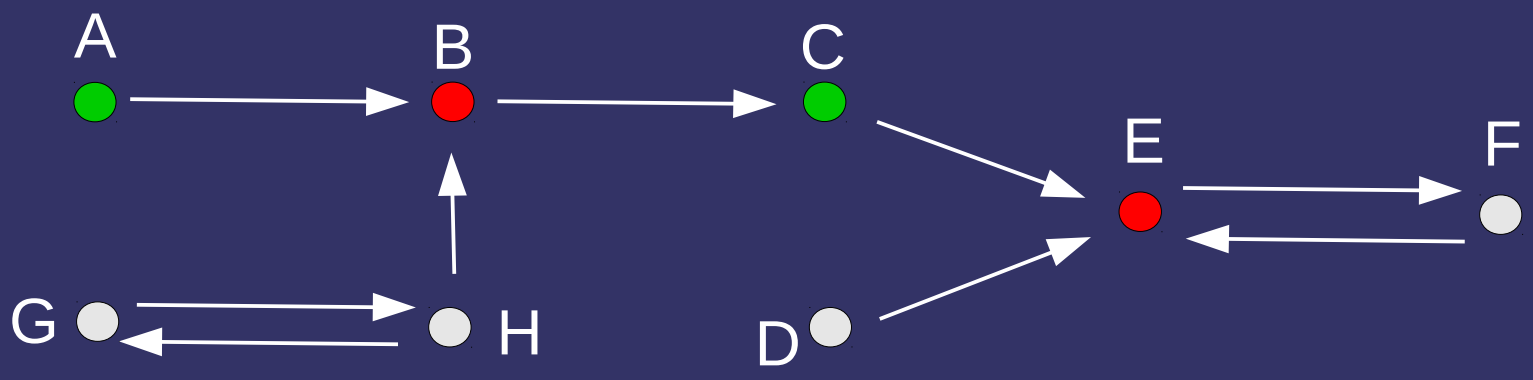
- in ⇔ all attackers out
- out ⇔ there is an attacker in
- undec ⇔ not all attackers out and no attacker in



Argumentation Theory: How to Evaluate Arguments

complete labelling:

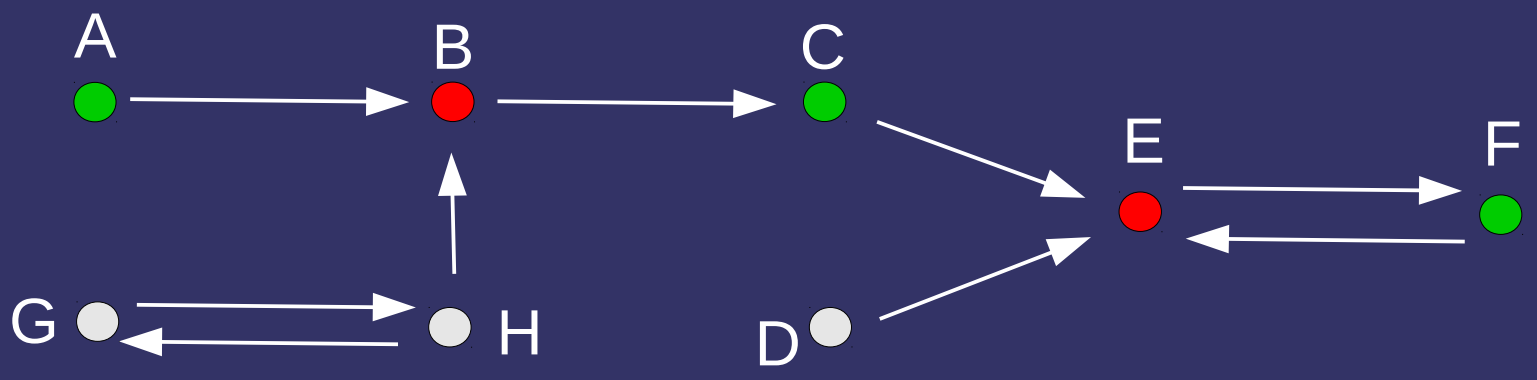
- in ⇔ all attackers out
- out ⇔ there is an attacker in
- undec ⇔ not all attackers out and no attacker in



Argumentation Theory: How to Evaluate Arguments

complete labelling:

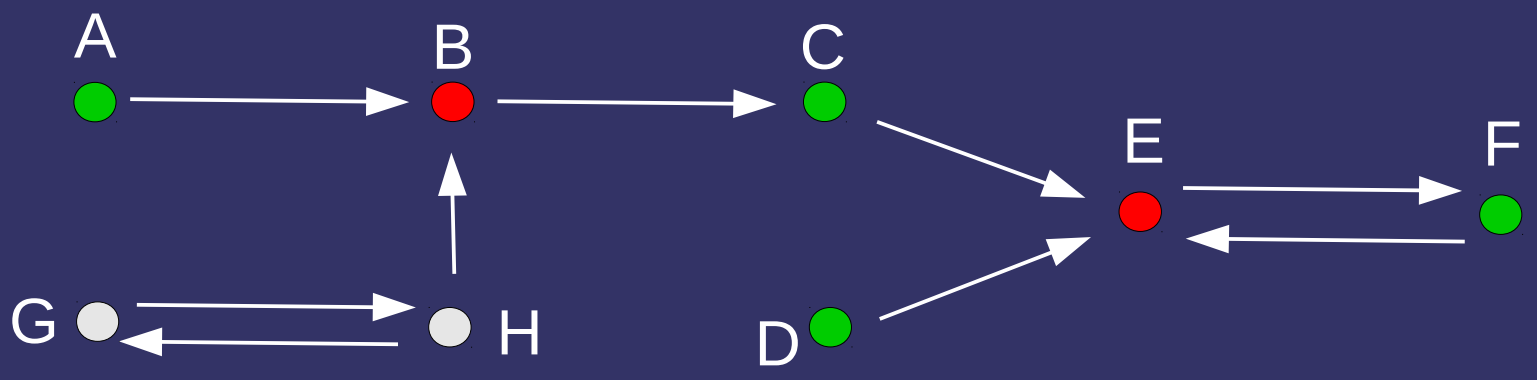
- in ⇔ all attackers out
- out ⇔ there is an attacker in
- undec ⇔ not all attackers out and no attacker in



Argumentation Theory: How to Evaluate Arguments

complete labelling:

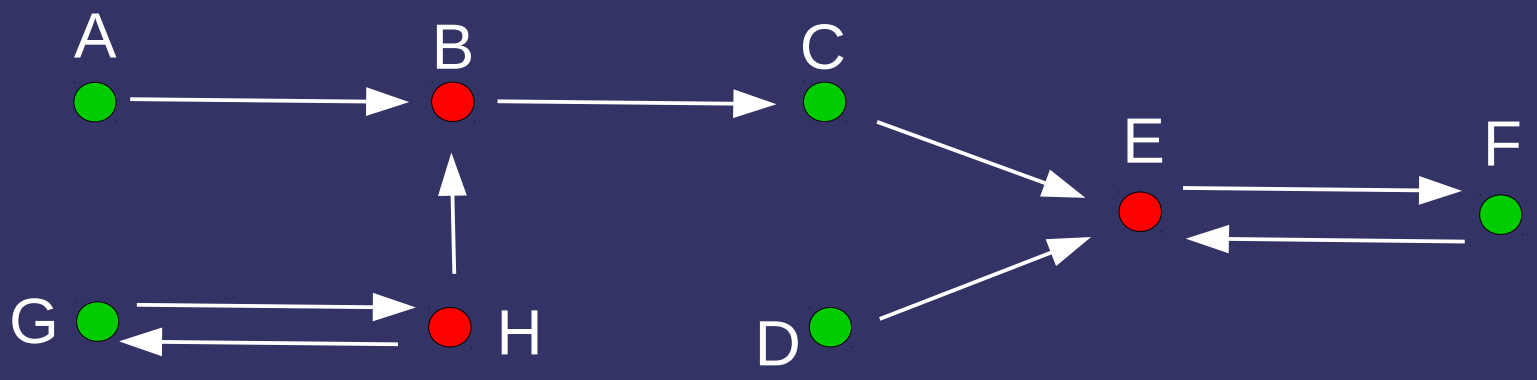
- in ⇔ all attackers out
- out ⇔ there is an attacker in
- undec ⇔ not all attackers out and no attacker in



Argumentation Theory: How to Evaluate Arguments

complete labelling:

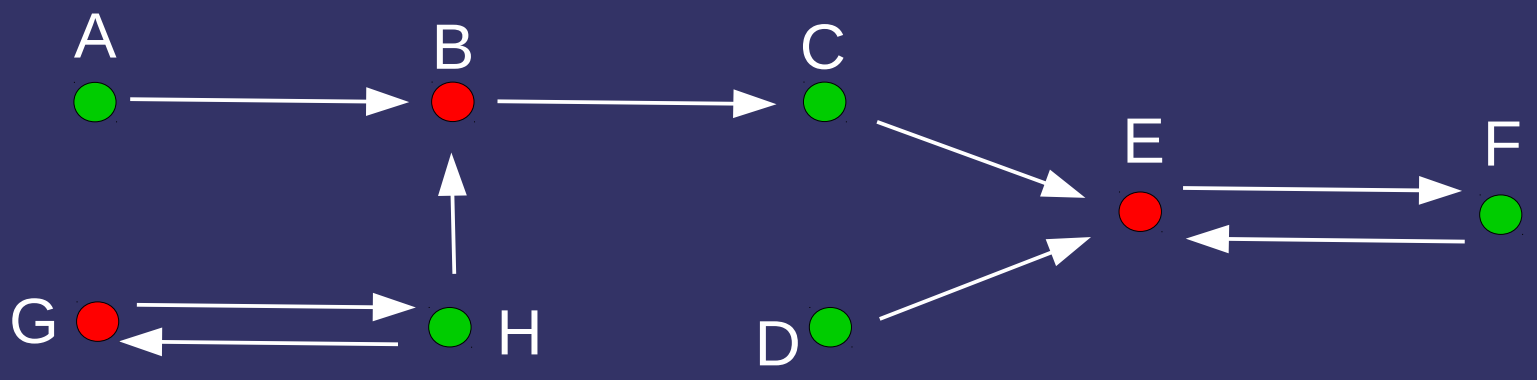
- in ⇔ all attackers out
- out ⇔ there is an attacker in
- undec ⇔ not all attackers out and no attacker in



Argumentation Theory: How to Evaluate Arguments

complete labelling:

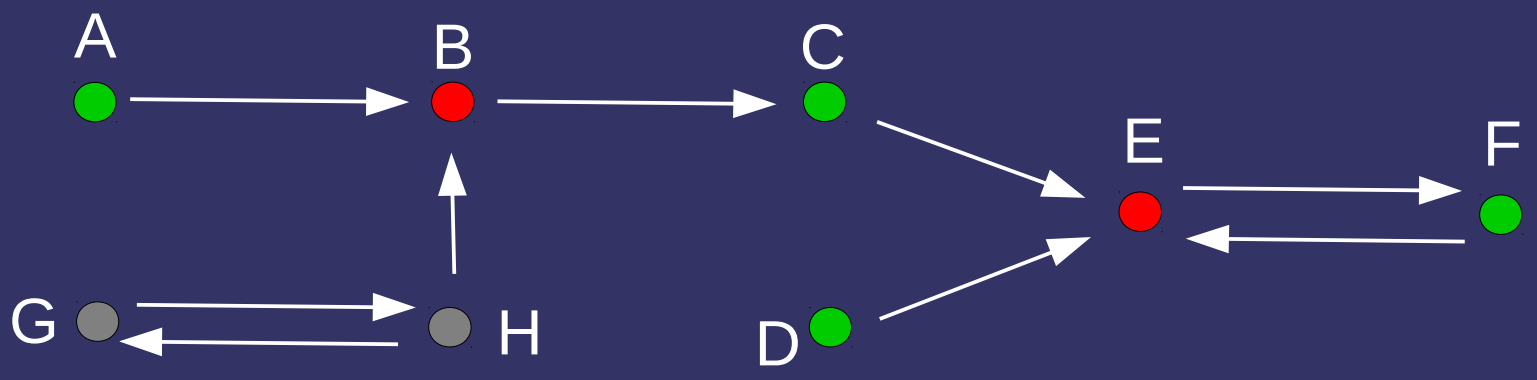
- in ⇔ all attackers out
- out ⇔ there is an attacker in
- undec ⇔ not all attackers out and no attacker in



Argumentation Theory: How to Evaluate Arguments

complete labelling:

- in ⇔ all attackers out
- out ⇔ there is an attacker in
- undec ⇔ not all attackers out and no attacker in



Argumentation Theory: How to Evaluate Arguments

complete labelling:

- in ⇔ all attackers out
- out ⇔ there is an attacker in
- undec ⇔ not all attackers out and no attacker in

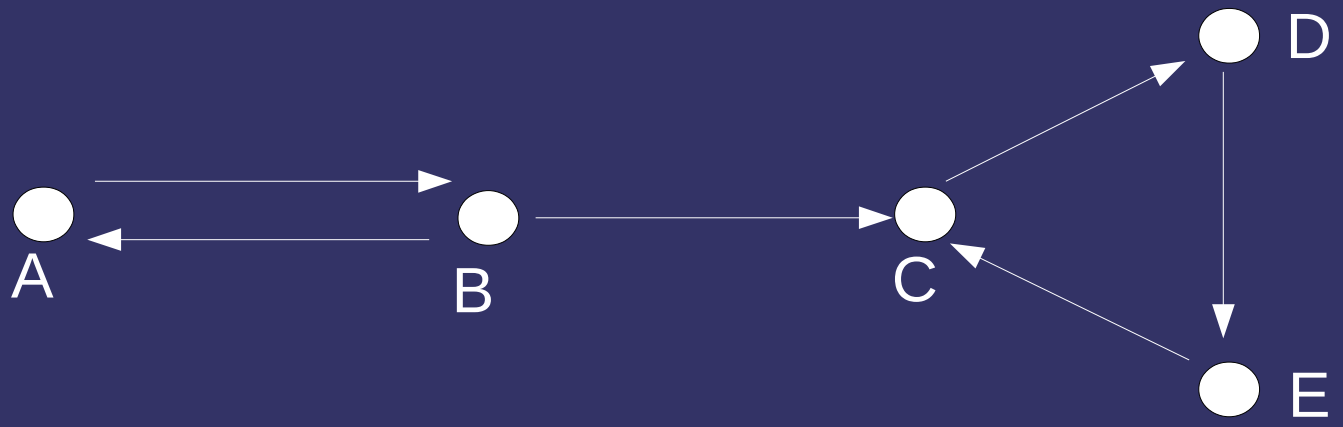
preferred labelling:
complete labelling with maximal in
(= complete labelling with maximal out)

grounded labelling:
complete labelling with maximal undec
(= complete labelling with minimal in/out)

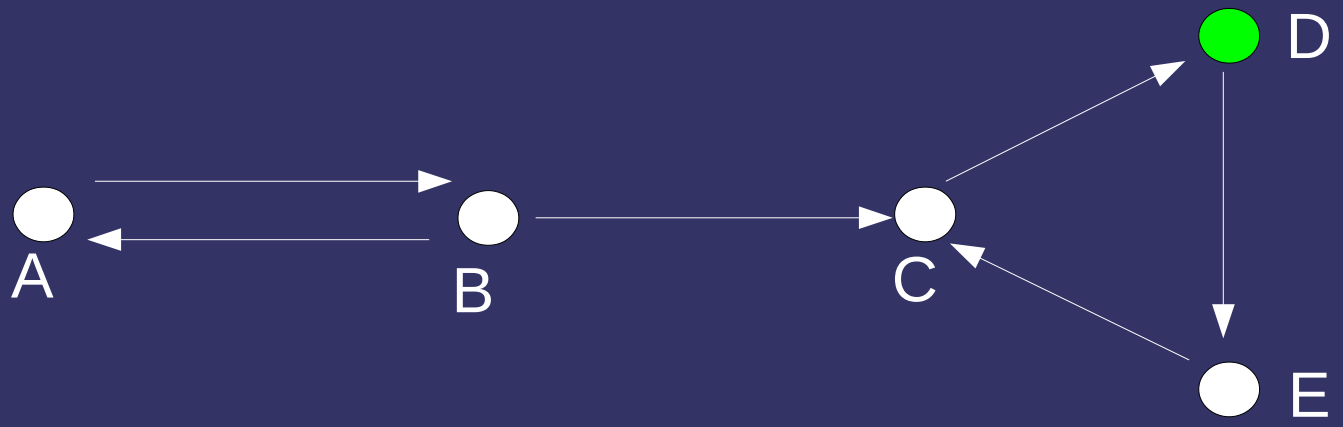
The Good News

- sound and complete discussion games have been stated for many types of labellings:
  - complete
  - preferred
  - grounded
  - stable
  - ideal
- We will now give an example of a discussion game for complete labellings...

Discussion Game for Complete Semantics

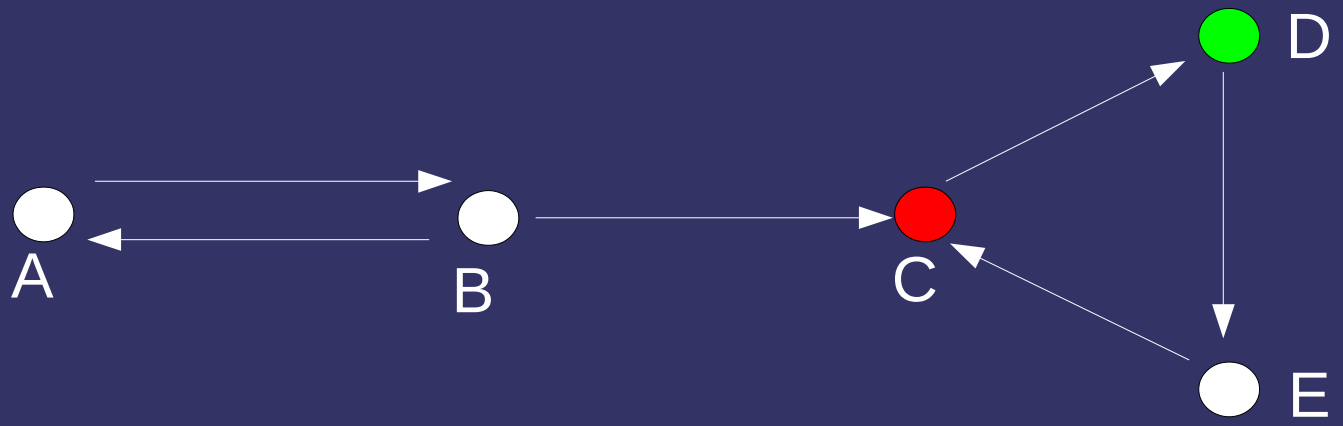


Discussion Game for Complete Semantics



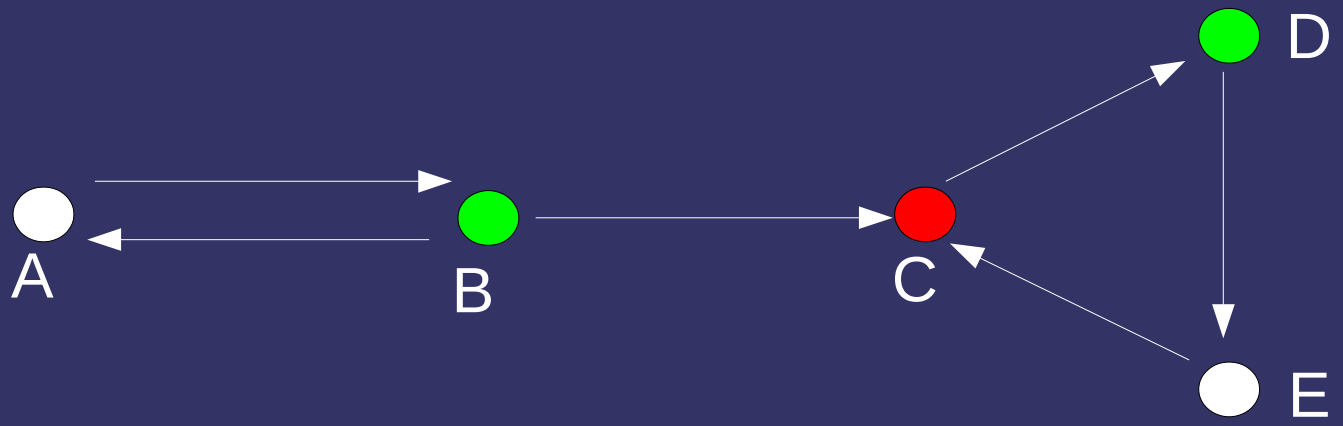
M: in(D) “I have a complete labelling in which D is labelled in.”

Discussion Game for Complete Semantics



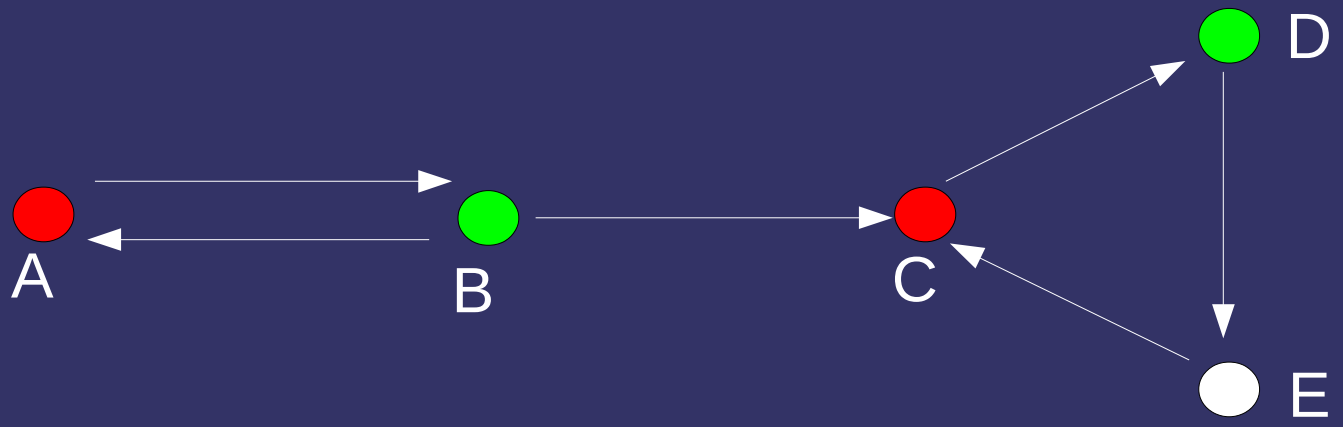
M: in(D) “I have a complete labelling in which D is labelled in.”
S: out(C) “But then in your labelling it must also be the case that D’s attacker C is labelled out. Based on which grounds?”

Discussion Game for Complete Semantics



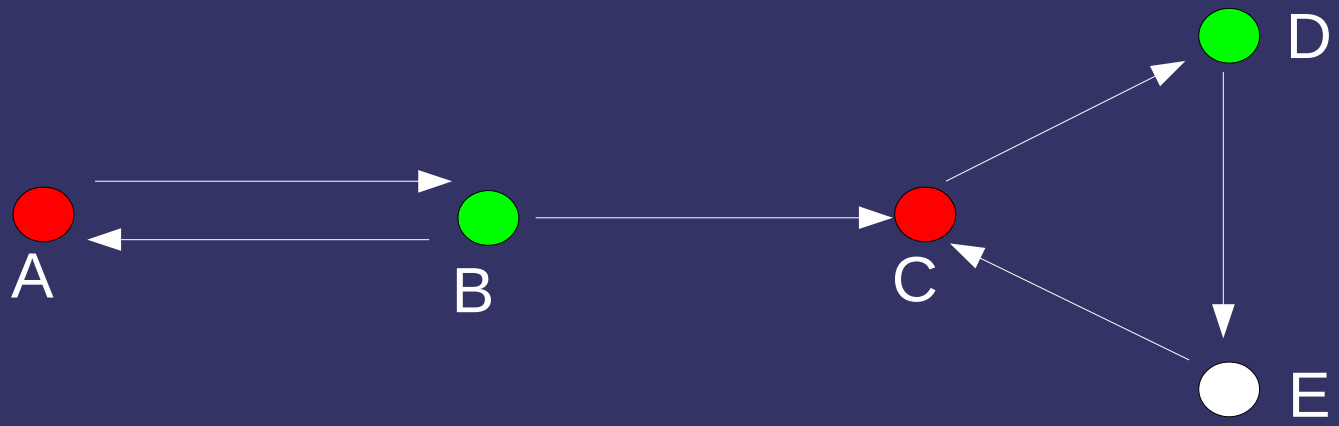
M: in(D) “I have a complete labelling in which D is labelled in.”
S: out(C) “But then in your labelling it must also be the case that D’s attacker C is labelled out. Based on which grounds?”
M: in(B) “C is labelled out because B is labelled in.”

Discussion Game for Complete Semantics



M: in(D) “I have a complete labelling in which D is labelled in.”
S: out(C) “But then in your labelling it must also be the case that D’s attacker C is labelled out. Based on which grounds?”
M: in(B) “C is labelled out because B is labelled in.”
S: out(A) “But then in your labelling it must also be the case that B’s attacker A is labelled out. Based on which grounds?”

Discussion Game for Complete Semantics



M: in(D) “I have a complete labelling in which D is labelled in.”
S: out(C) “But then in your labelling it must also be the case that D’s attacker C is labelled out. Based on which grounds?”
M: in(B) “C is labelled out because B is labelled in.”
S: out(A) “But then in your labelling it must also be the case that B’s attacker A is labelled out. Based on which grounds?”
M: in(B) “A is labelled out because B is labelled in.”

Discussion Game for Complete Semantics

M: in(D) “I have a complete labelling in which D is labelled in.”
S: out(C) “But then in your labelling it must also be the case that D’s attacker C is labelled out. Based on which grounds?”
M: in(B) “C is labelled out because B is labelled in.”
S: out(A) “But then in your labelling it must also be the case that B’s attacker A is labelled out. Based on which grounds?”
M: in(B) “A is labelled out because B is labelled in.”

Discussion Game for Complete Semantics

M: in(D) “I have a complete labelling in which D is labelled in.”
S: out(C) “But then in your labelling it must also be the case that D’s attacker C is labelled out. Based on which grounds?”
M: in(B) “C is labelled out because B is labelled in.”
S: out(A) “But then in your labelling it must also be the case that B’s attacker A is labelled out. Based on which grounds?”
M: in(B) “A is labelled out because B is labelled in.”

Discussion Game for Complete Semantics

M: in(D) “I have a complete labelling in which D is labelled in.”
S: out(C) “But then in your labelling it must also be the case that D’s attacker C is labelled out. Based on which grounds?”
M: in(B) “C is labelled out because B is labelled in.”
S: out(A) “But then in your labelling it must also be the case that B’s attacker A is labelled out. Based on which grounds?”
M: in(B) “A is labelled out because B is labelled in.”

(1) Each move of M (except the first) contains an attacker of the directly preceding move of S.

Discussion Game for Complete Semantics

M: in(D) “I have a complete labelling in which D is labelled in.”
S: out(C) “But then in your labelling it must also be the case that D’s attacker C is labelled out. Based on which grounds?”
M: in(B) “C is labelled out because B is labelled in.”
S: out(A) “But then in your labelling it must also be the case that B’s attacker A is labelled out. Based on which grounds?”
M: in(B) “A is labelled out because B is labelled in.”

(2) Each move of S
contains an attacker of <u>some</u> previous move of M.

Discussion Game for Complete Semantics

M: in(D) “I have a complete labelling in which D is labelled in.”
S: out(C) “But then in your labelling it must also be the case that D’s attacker C is labelled out. Based on which grounds?”
M: in(B) “C is labelled out because B is labelled in.”
S: out(A) “But then in your labelling it must also be the case that B’s attacker A is labelled out. Based on which grounds?”
M: in(B) “A is labelled out because B is labelled in.”

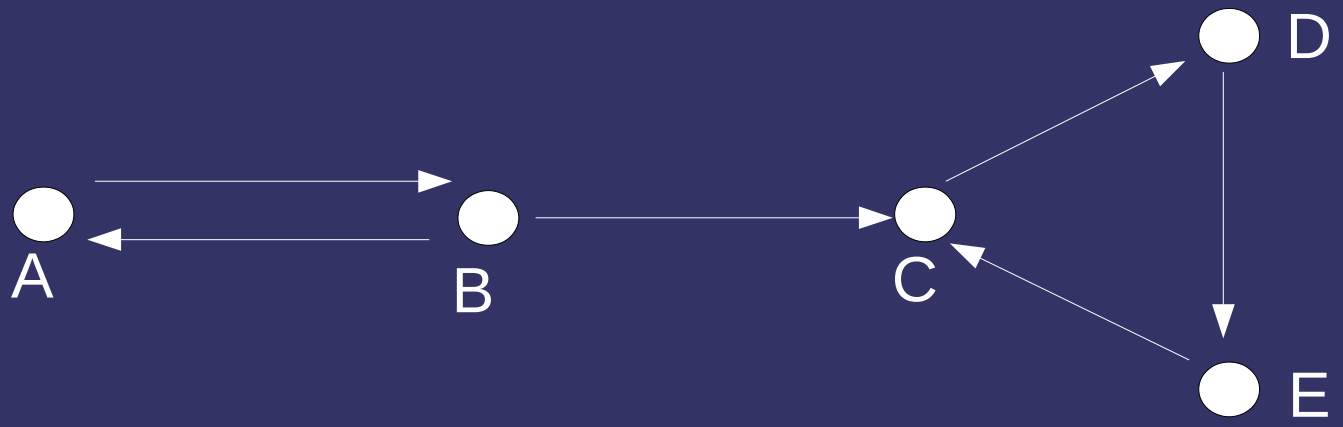
(3) S is not allowed to repeat his moves.

Discussion Game for Complete Semantics

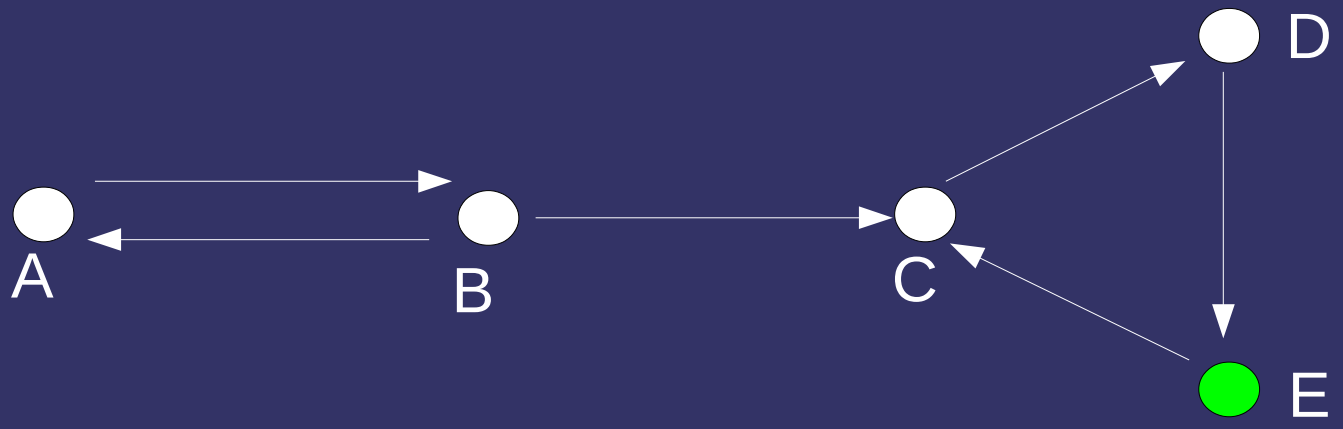
M: in(D) “I have a complete labelling in which D is labelled in.”
S: out(C) “But then in your labelling it must also be the case that D’s attacker C is labelled out. Based on which grounds?”
M: in(B) “C is labelled out because B is labelled in.”
S: out(A) “But then in your labelling it must also be the case that B’s attacker A is labelled out. Based on which grounds?”
M: in(B) “A is labelled out because B is labelled in.”

(4) M is allowed to repeat his moves.

Discussion Game for Complete Semantics

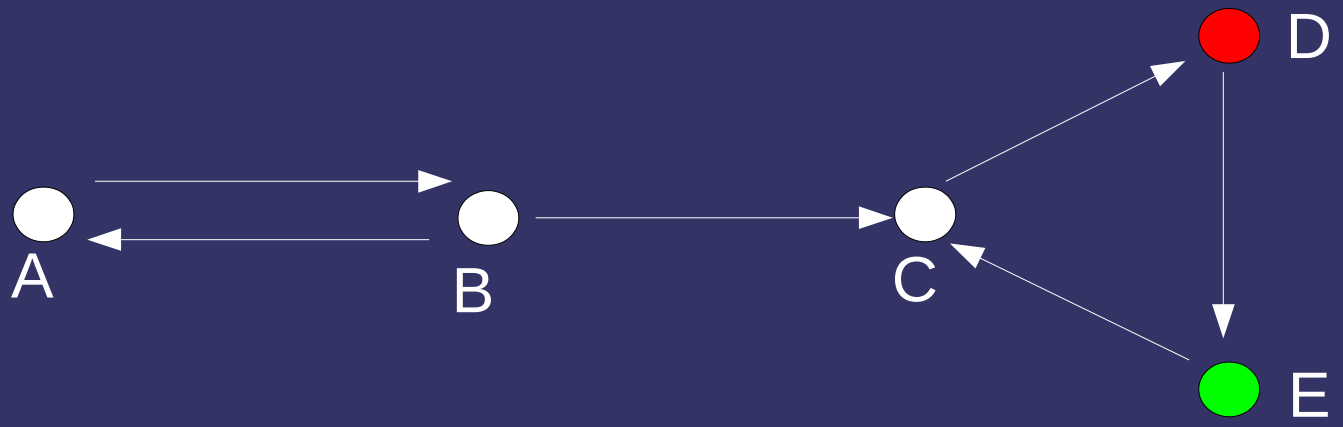


Discussion Game for Complete Semantics



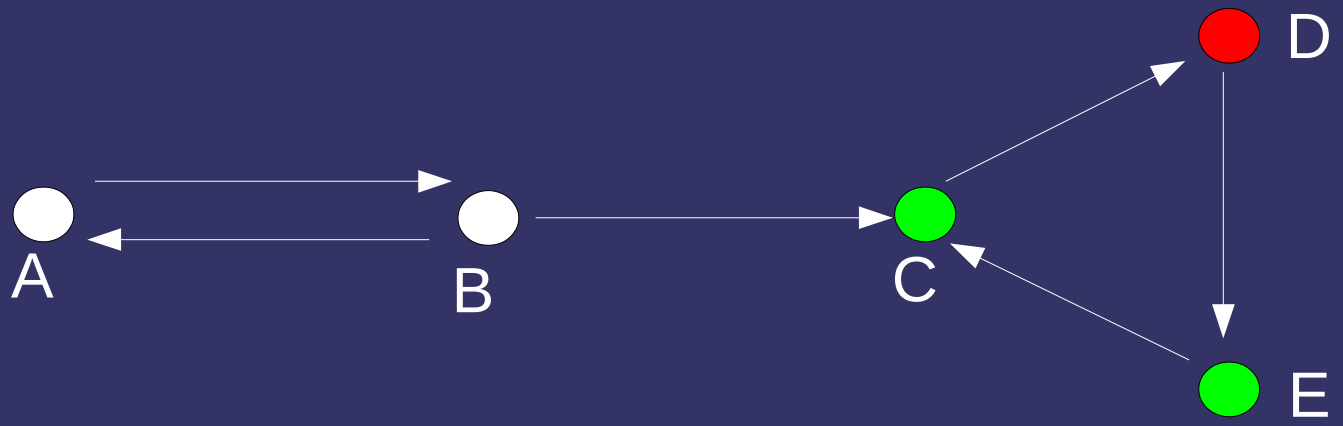
M: in(E) “I have a complete labelling in which E is labelled in.”

Discussion Game for Complete Semantics



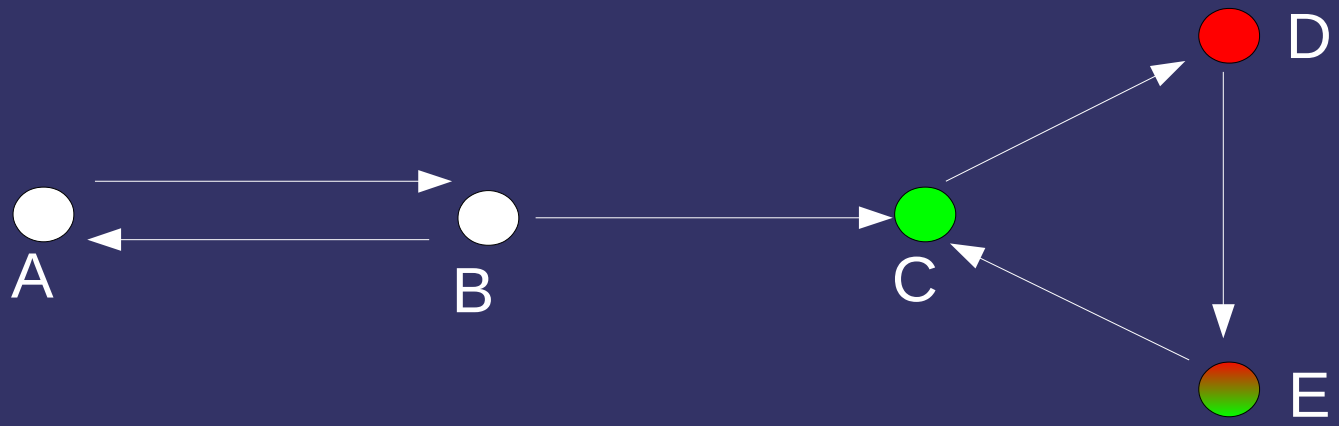
M: in(E) “I have a complete labelling in which E is labelled in.”
S: out(D) “But then in your labelling it must also be the case that E's attacker D is labelled out. Based on which grounds?”

Discussion Game for Complete Semantics



M: in(E) “I have a complete labelling in which E is labelled in.”
S: out(D) “But then in your labelling it must also be the case that E's attacker D is labelled out. Based on which grounds?”
M: in(C) “D is labelled out because C is labelled in.”

Discussion Game for Complete Semantics



M: in(E) “I have a complete labelling in which E is labelled in.”
S: out(D) “But then in your labelling it must also be the case that E's attacker D is labelled out. Based on which grounds?”
M: in(C) “D is labelled out because C is labelled in.”
S: out(E) “But then in your labelling it must also be the case that C's attacker E is labelled out. This *contradicts* with your earlier claim that E is labelled in.”

Discussion Game for Complete Semantics

M: in(E) “I have a complete labelling in which E is labelled in.”
S: out(D) “But then in your labelling it must also be the case that E's attacker D is labelled out. Based on which grounds?”
M: in(C) “D is labelled out because C is labelled in.”
S: out(E) “But then in your labelling it must also be the case that C's attacker E is labelled out. This *contradicts* with your earlier claim that E is labelled in.”

Discussion Game for Complete Semantics

M: in(E) “I have a complete labelling in which E is labelled in.”

S: out(D) “But then in your labelling it must also be the case that E's attacker D is labelled out. Based on which grounds?”

M: in(C) “D is labelled out because C is labelled in.”

S: out(E) “But then in your labelling it must also be the case that C's attacker E is labelled out. This *contradicts* with your earlier claim that E is labelled in.”

Discussion Game for Complete Semantics

M: in(E) “I have a complete labelling in which E is labelled in.”
S: out(D) “But then in your labelling it must also be the case that E's attacker D is labelled out. Based on which grounds?”
M: in(C) “D is labelled out because C is labelled in.”
S: out(E) “But then in your labelling it must also be the case that C's attacker E is labelled out. This *contradicts* with your earlier claim that E is labelled in.”

(5) If S uses an argument previously used by M, then S wins the discussion.

Discussion Game for Complete Semantics

M: in(E) “I have a complete labelling in which E is labelled in.”
S: out(D) “But then in your labelling it must also be the case that E's attacker D is labelled out. Based on which grounds?”
M: in(C) “D is labelled out because C is labelled in.”
S: out(E) “But then in your labelling it must also be the case that C's attacker E is labelled out. This *contradicts* with your earlier claim that E is labelled in.”

(6) If M uses an argument previously used by S, then S wins the discussion.

Discussion Game for Complete Semantics

M: in(E) “I have a complete labelling in which E is labelled in.”
S: out(D) “But then in your labelling it must also be the case that E's attacker D is labelled out. Based on which grounds?”
M: in(C) “D is labelled out because C is labelled in.”
S: out(E) “But then in your labelling it must also be the case that C's attacker E is labelled out. This *contradicts* with your earlier claim that E is labelled in.”

(7) If M cannot make a move anymore, then S wins the discussion.

Discussion Game for Complete Semantics

M: in(E) “I have a complete labelling in which E is labelled in.”
S: out(D) “But then in your labelling it must also be the case that E's attacker D is labelled out. Based on which grounds?”
M: in(C) “D is labelled out because C is labelled in.”
S: out(E) “But then in your labelling it must also be the case that C's attacker E is labelled out. This *contradicts* with your earlier claim that E is labelled in.”

(8) If S cannot make a move anymore, then M wins the discussion.

Discussion Game for Complete Semantics

(1) Each move of M (except the first)
contains an attacker of the directly preceding move of S.
(2) Each move of S
contains an attacker of some previous move of M.
(3) S is not allowed to repeat his moves.
(4) M is allowed to repeat his moves.
(5) If S uses an argument previously used by M,
then S wins the discussion.
(6) If M uses an argument previously used by S,
then S wins the discussion.
(7) If M cannot make a move anymore,
then S wins the discussion.
(8) If S cannot make a move anymore,
then M wins the discussion.

Constructing the Graph
What Attacks What

$((d) \Rightarrow e) \Rightarrow \neg c$

$((a) \Rightarrow b) \Rightarrow c$

strict rule (→):

"from ... it <u>always</u> follows that..."

defeasible rule (⇒):

"from ... it <u>usually</u> follows that..."

Constructing the Graph
What Attacks What

$((d) \Rightarrow e) \Rightarrow \neg c$

$((a) \rightarrow b) \rightarrow c$

strict rule (→):

“from ... it <u>always</u> follows that...”

defeasible rule (⇒):

“from ... it <u>usually</u> follows that...”

Constructing the Graph
What Attacks What

$((d) \rightarrow e) \Rightarrow \neg c$

$((a) \Rightarrow b) \rightarrow c$

strict rule ($\rightarrow$):

"from ... it <u>always</u> follows that..."

defeasible rule ($\Rightarrow$):

"from ... it <u>usually</u> follows that..."

Some Terminology

- unrestricted rebut: *check if <u>any previous rule</u>*
  (of the attacked conclusion) is defeasible

  so ((d) → e) ⇒ ¬c attacks
  ((a) ⇒ b) → c
  but not ((a) → b) → c

- restricted rebut: *check if the <u>last rule</u>*
  (of the attacked conclusion) is defeasible

  so ((d) → e) ⇒ ¬c attacks
  ((a) → b) ⇒ c
  but not ((a) ⇒ b) → c

A Sample Discussion

John: *“Bob will go to both AAMAS and IJCAI this year,*
as he has papers accepted at each of these conferences.”
Mary: *“That won't be possible,*
as his budget only allows one foreign trip.”

A Sample Discussion

S = { →paperA; →paperI; →budget;

D = { paperA⇒goA; paperI⇒goI; budget⇒¬goboth) }

John: *“Bob will go to both AAMAS and IJCAI this year,*
as he has papers accepted at each of these conferences.”
Mary: *“That won't be possible,*
as his budget only allows one foreign trip.”

A Sample Discussion

S = { →paperA; →paperI; →budget;
goA,goI→goboth; goA,¬goboth→¬goI; goI,¬goboth→¬goA }
D = { paperA⇒goA; paperI⇒goI; budget⇒¬goboth) }

John: *“Bob will go to both AAMAS and IJCAI this year,
as he has papers accepted at each of these conferences.”*
Mary: *“That won't be possible,
as his budget only allows one foreign trip.”*

A Sample Discussion

S = { →paperA; →paperI; →budget;
goA,goI→goboth; goA,¬goboth→¬goI; goI,¬goboth→¬goA }
D = { paperA⇒goA; paperI⇒goI; budget⇒¬goboth) }

John: ((→paperA)⇒goA), ((→paperI)⇒goI) → goboth
Mary: (→budget)⇒¬goboth

John: *“Bob will go to both AAMAS and IJCAI this year,
as he has papers accepted at each of these conferences.”*
Mary: *“That won't be possible,
as his budget only allows one foreign trip.”*

A Sample Discussion

S = { →paperA; →paperI; →budget;
goA,goI→goboth; goA,¬goboth→¬goI; goI,¬goboth→¬goA }
D = { paperA⇒goA; paperI⇒goI; budget⇒¬goboth) }

John: ((→paperA)⇒goA), ((→paperI)⇒goI) → goboth
Mary: ((→paperA)⇒goA), ((→budget)⇒¬goboth) → ¬goI

John: *“Bob will go to both AAMAS and IJCAI this year,
as he has papers accepted at both conferences.”*
Mary: *“Bob goes to AAMAS, so he cannot go to IJCAI,
as his budget only allows one foreign trip.”*

A Sample Discussion

S = { →paperA; →paperI; →budget;
goA,goI→goboth; goA,¬goboth→¬goI; goI,¬goboth→¬goA }
D = { paperA⇒goA; paperI⇒goI; budget⇒¬goboth) }

John: ((→paperA)⇒goA), ((→paperI)⇒goI) → goboth
Mary: ((→paperA)⇒goI), ((→budget)⇒¬goboth) → ¬goA

John: *“Bob will go to both AAMAS and IJCAI this year,
as he has papers accepted at both conferences.”*
Mary: *“Bob goes to IJCAI, so he cannot go to AAMAS,
as his budget only allows one foreign trip.”*

Why Restricted Rebut

S = { w1; w2; w3;
D = { w1 ⇒ b1;   w2 ⇒ b2;   w3 ⇒ b3 }

Why Restricted Rebut

S = { w1; w2; w3; b2,b3→¬b1; b1,b3→¬b2; b1,b2→¬b3 }

D = { w1 ⇒ b1;   w2 ⇒ b2;   w3 ⇒ b3 }

Why Restricted Rebut

S = { w1; w2; w3; b2,b3→¬b1; b1,b3→¬b2; b1,b2→¬b3 }
D = { w1 ⇒ b1;   w2 ⇒ b2;   w3 ⇒ b3 }

A_1: (w1) ⇒ b1
A_2: (w2) ⇒ b2
A_3: (w3) ⇒ b3

Why Restricted Rebut

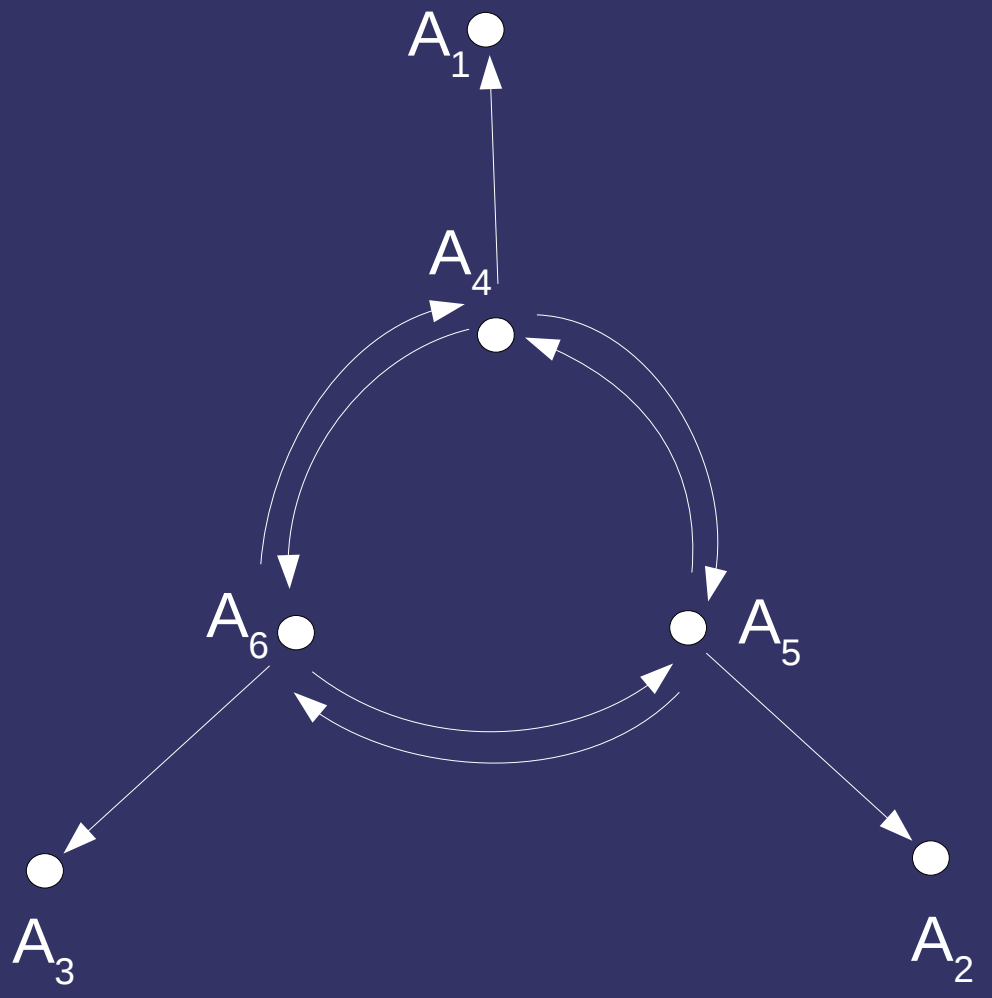
S = { w1; w2; w3; b2,b3→¬b1; b1,b3→¬b2; b1,b2→¬b3 }
D = { w1 ⇒ b1; w2 ⇒ b2; w3 ⇒ b3 }

A_1: (w1) ⇒ b1
A_2: (w2) ⇒ b2
A_3: (w3) ⇒ b3
A_4: A_2, A_3 → ¬b1
A_5: A_1, A_3 → ¬b2
A_6: A_1, A_2 → ¬b3

Why Restricted Rebut

S = { w1; w2; w3; b2,b3→¬b1; b1,b3→¬b2; b1,b2→¬b3 }
D = { w1 ⇒ b1; w2 ⇒ b2; w3 ⇒ b3 }

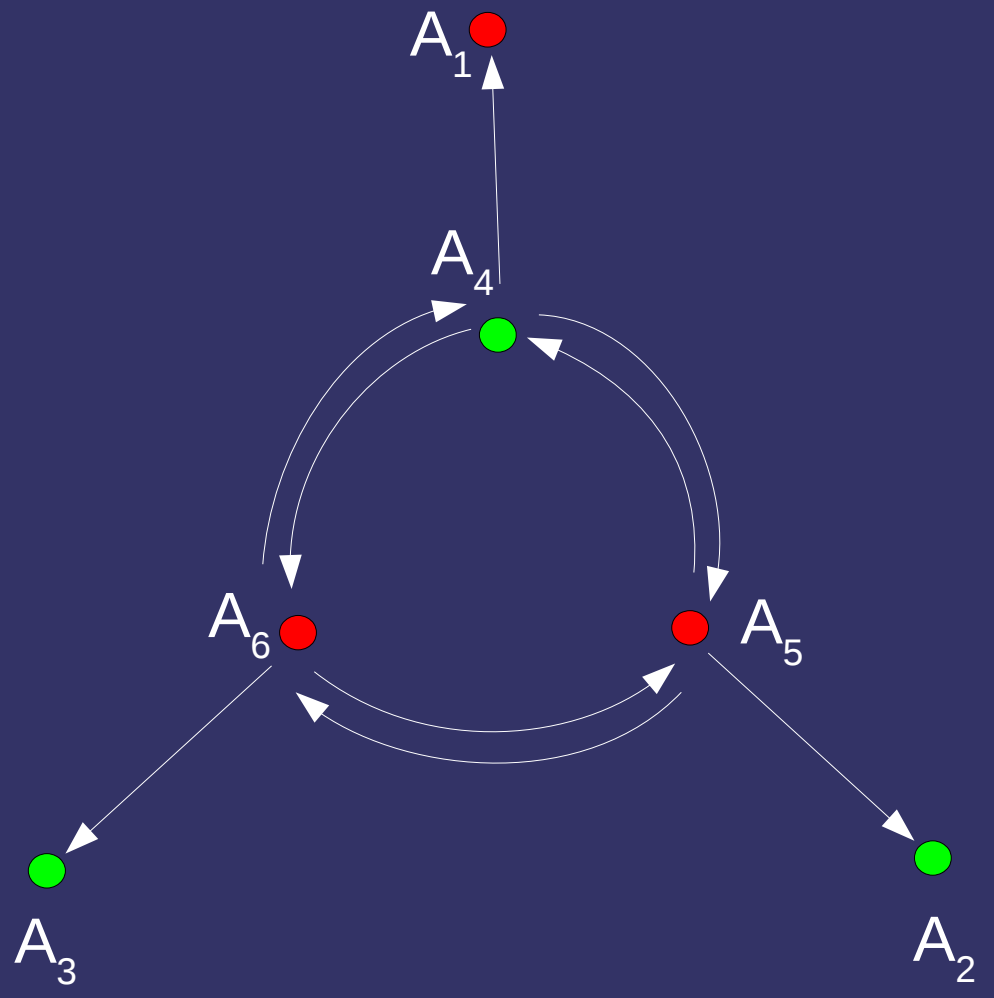
A_1: (w1) ⇒ b1
A_2: (w2) ⇒ b2
A_3: (w3) ⇒ b3
A_4: A_2, A_3 → ¬b1
A_5: A_1, A_3 → ¬b2
A_6: A_1, A_2 → ¬b3



Why Restricted Rebut

S = { w1; w2; w3; b2,b3→¬b1; b1,b3→¬b2; b1,b2→¬b3 }
D = { w1 ⇒ b1; w2 ⇒ b2; w3 ⇒ b3 }

A_1: (w1) ⇒ b1
A_2: (w2) ⇒ b2
A_3: (w3) ⇒ b3
A_4: A_2, A_3 → ¬b1
A_5: A_1, A_3 → ¬b2
A_6: A_1, A_2 → ¬b3

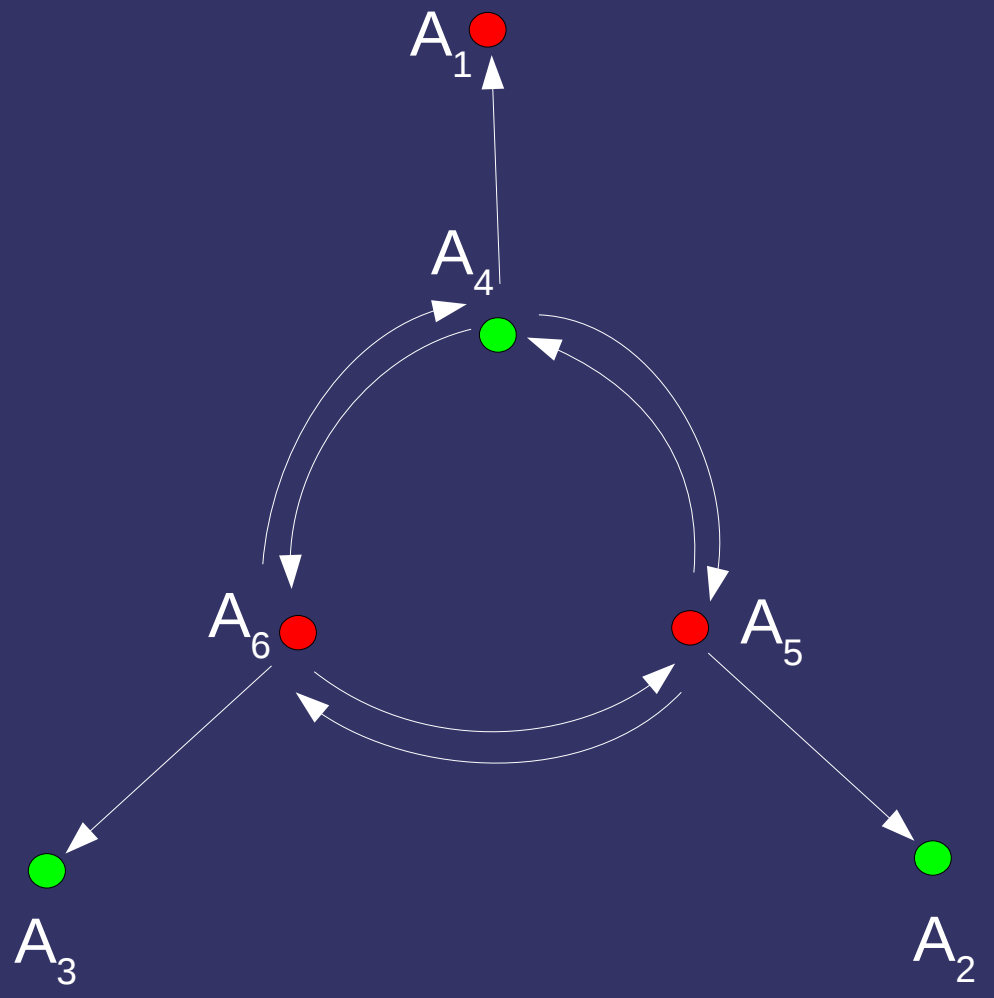


Why Restricted Rebut

S = { w1; w2; w3; b2,b3→¬b1; b1,b3→¬b2; b1,b2→¬b3 }
D = { w1 ⇒ b1; w2 ⇒ b2; w3 ⇒ b3 }

A_1: (w1) ⇒ b1
A_2: (w2) ⇒ b2
A_3: (w3) ⇒ b3
A_4: A_2, A_3 → ¬b1
A_5: A_1, A_3 → ¬b2
A_6: A_1, A_2 → ¬b3

b1 b2 b3 ¬b1 ¬b2 ¬b3

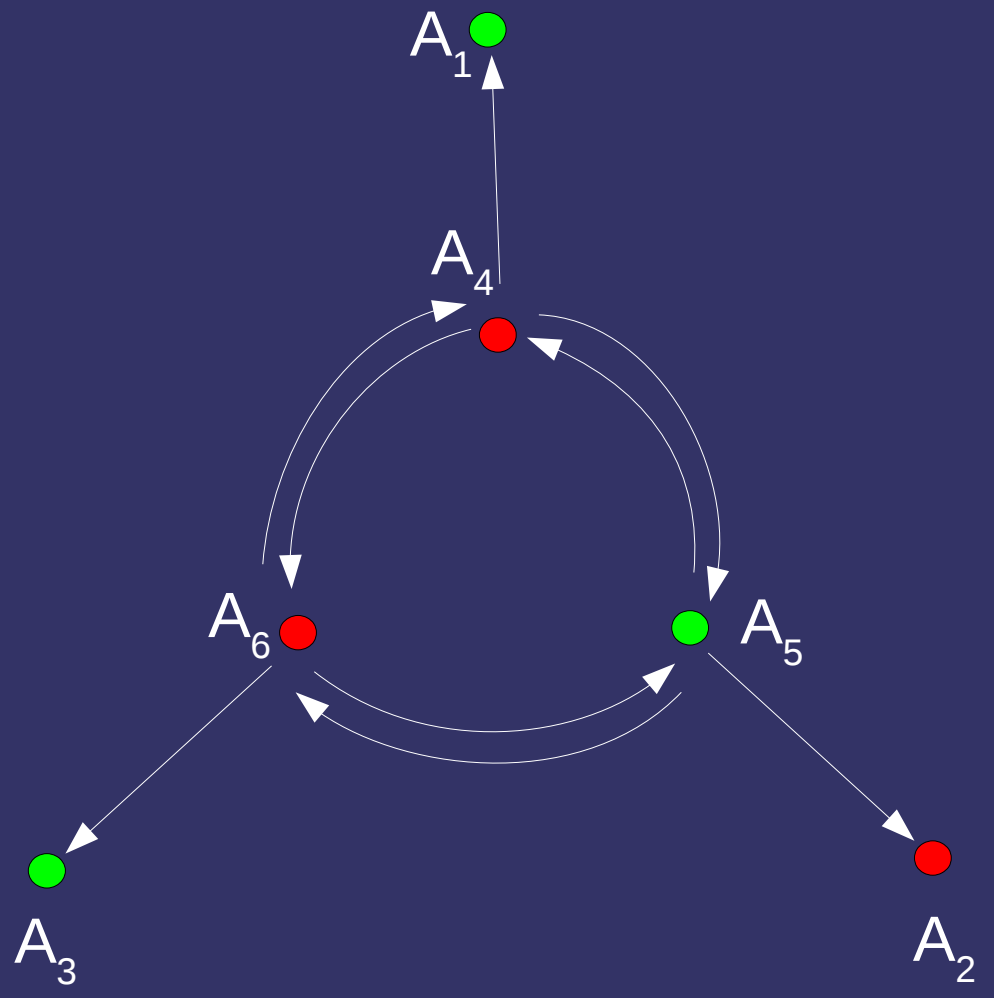


Why Restricted Rebut

S = { w1; w2; w3; b2,b3→¬b1; b1,b3→¬b2; b1,b2→¬b3 }
D = { w1 ⇒ b1; w2 ⇒ b2; w3 ⇒ b3 }

A_1: (w1) ⇒ b1
A_2: (w2) ⇒ b2
A_3: (w3) ⇒ b3
A_4: A_2, A_3 → ¬b1
A_5: A_1, A_3 → ¬b2
A_6: A_1, A_2 → ¬b3

b1 b2 b3 ¬b1 ¬b2 ¬b3

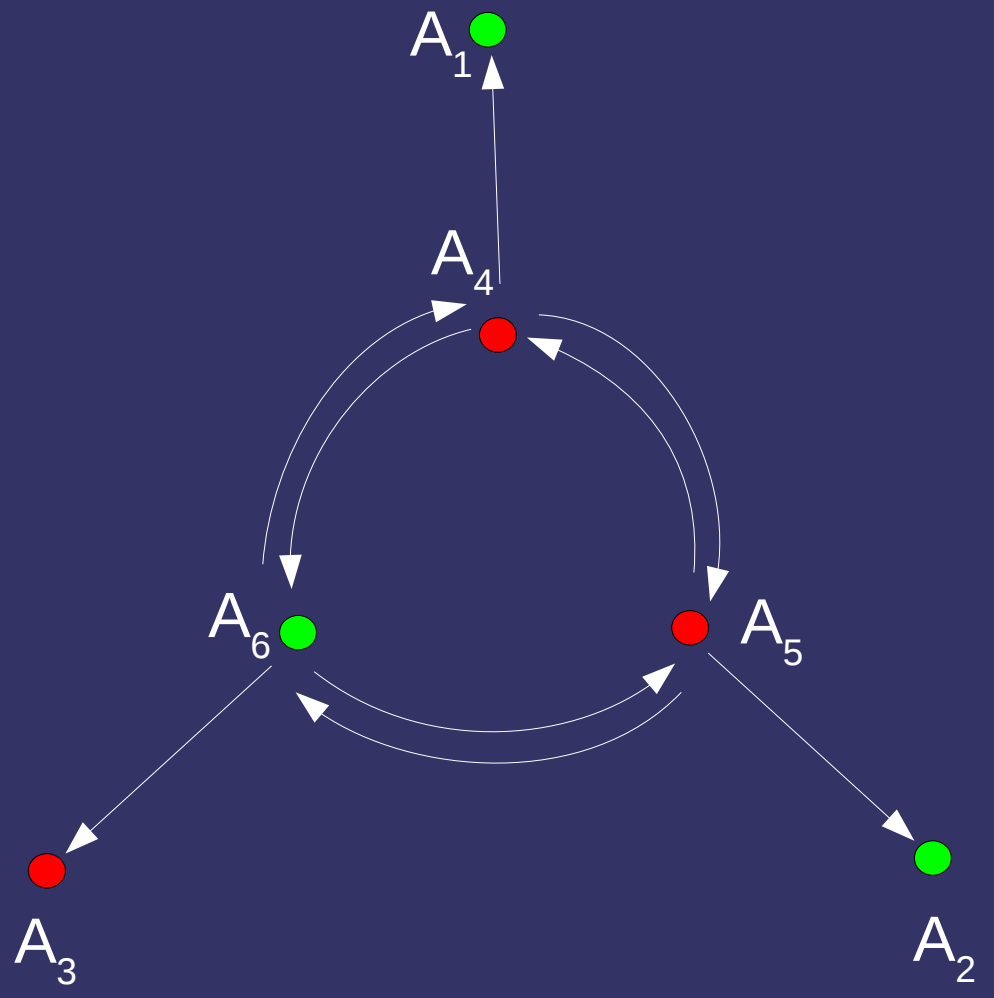


Why Restricted Rebut

S = { w1; w2; w3; b2,b3→¬b1; b1,b3→¬b2; b1,b2→¬b3 }
D = { w1 ⇒ b1; w2 ⇒ b2; w3 ⇒ b3 }

A_1: (w1) ⇒ b1
A_2: (w2) ⇒ b2
A_3: (w3) ⇒ b3
A_4: A_2, A_3 → ¬b1
A_5: A_1, A_3 → ¬b2
A_6: A_1, A_2 → ¬b3

b1 b2 b3 ¬b1 ¬b2 ¬b3

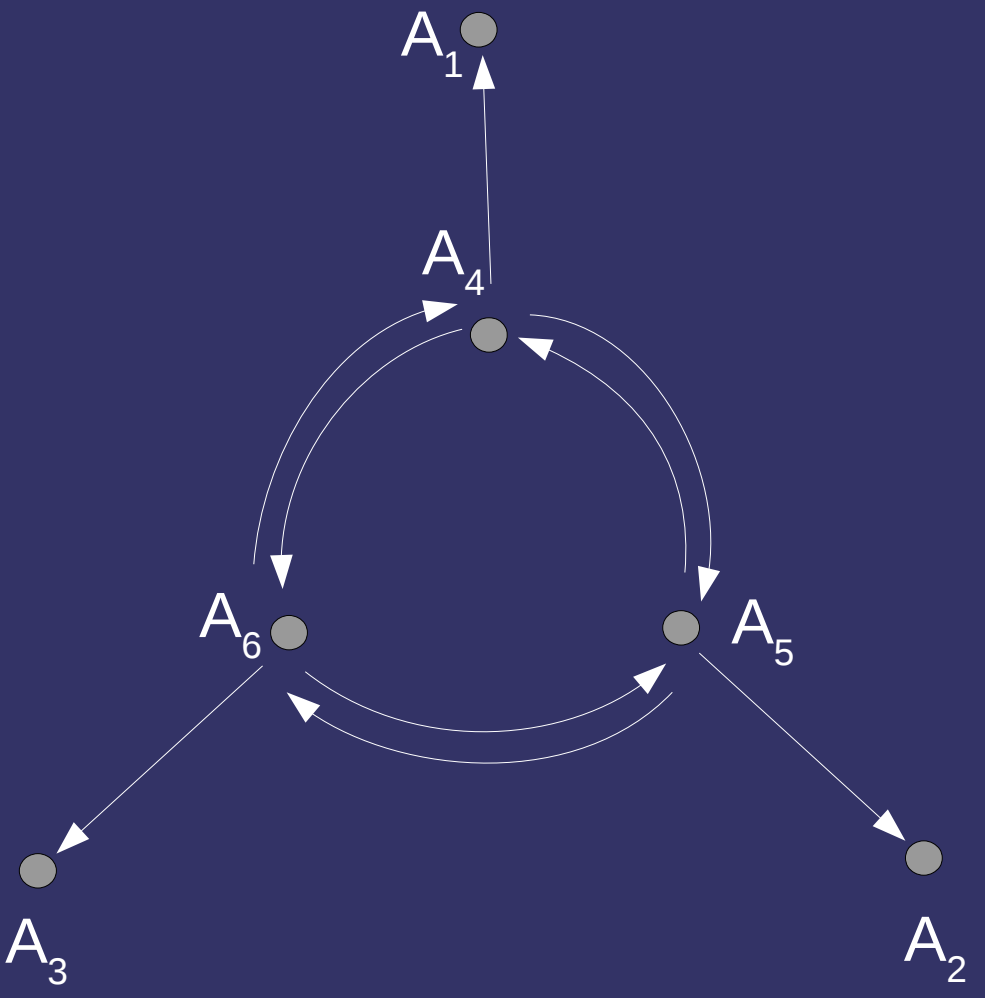


Why Restricted Rebut

S = { w1; w2; w3; b2,b3→¬b1; b1,b3→¬b2; b1,b2→¬b3 }
D = { w1 ⇒ b1;   w2 ⇒ b2;   w3 ⇒ b3 }

A_1: (w1) ⇒ b1
A_2: (w2) ⇒ b2
A_3: (w3) ⇒ b3
A_4: A_2, A_3 → ¬b1
A_5: A_1, A_3 → ¬b2
A_6: A_1, A_2 → ¬b3

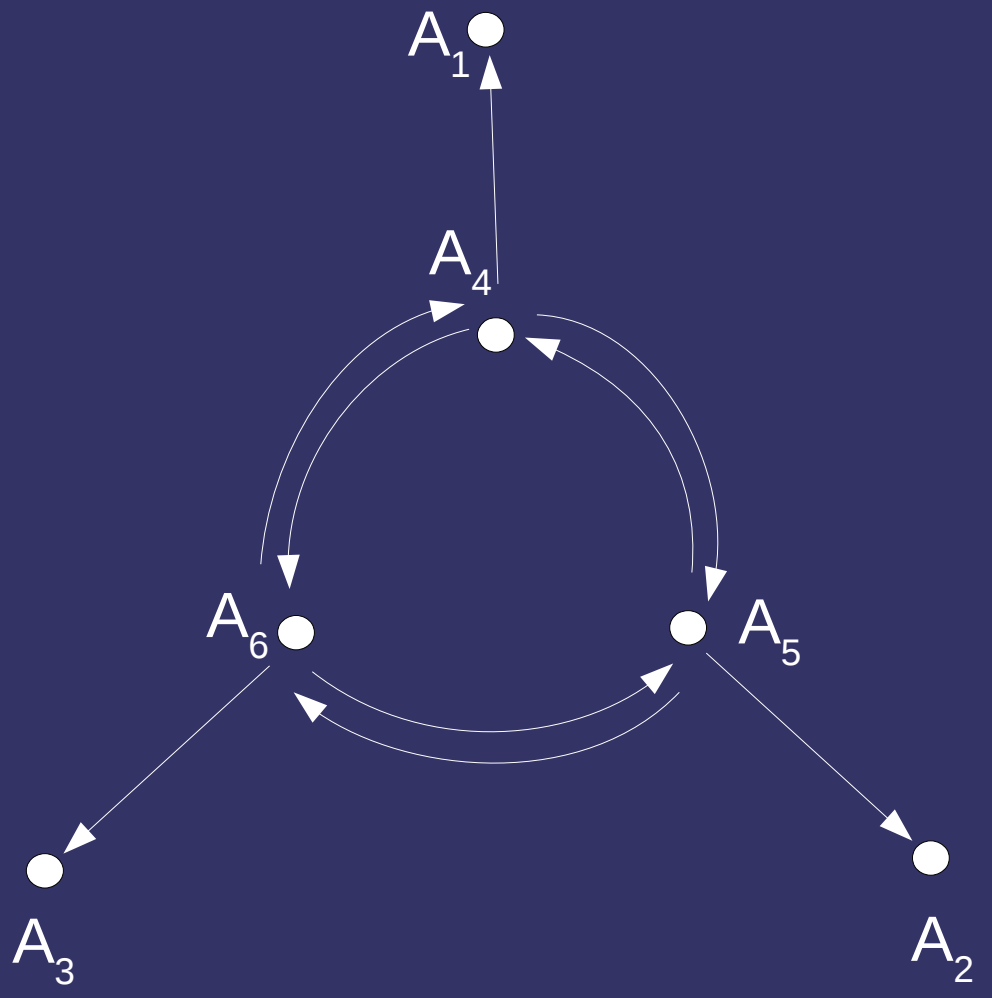
b1 b2 b3   ¬b1 ¬b2 ¬b3



Why Restricted Rebut

S = { w1; w2; w3; b2,b3→¬b1; b1,b3→¬b2; b1,b2→¬b3 }
D = { w1 ⇒ b1; w2 ⇒ b2; w3 ⇒ b3 }

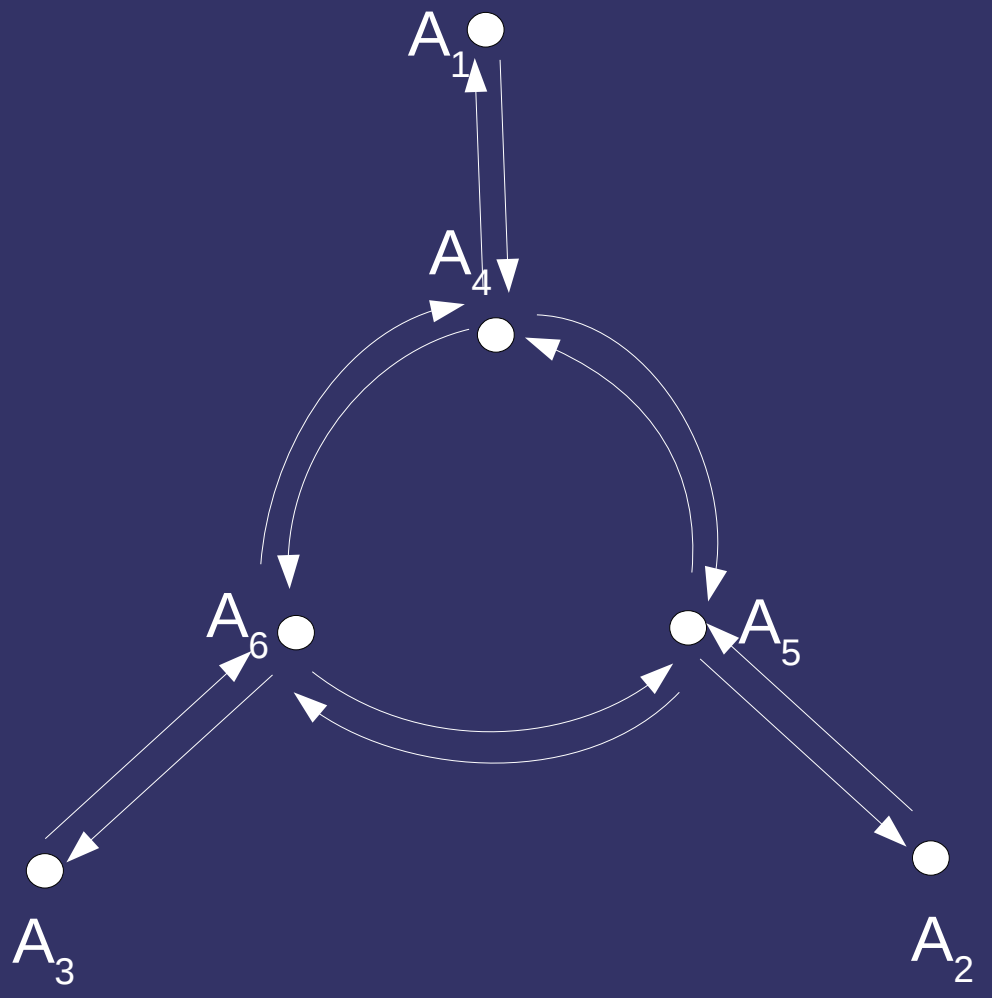
A_1: (w1) ⇒ b1
A_2: (w2) ⇒ b2
A_3: (w3) ⇒ b3
A_4: A_2, A_3 → ¬b1
A_5: A_1, A_3 → ¬b2
A_6: A_1, A_2 → ¬b3



Why Restricted Rebut

S = { w1; w2; w3; b2,b3→¬b1; b1,b3→¬b2; b1,b2→¬b3 }
D = { w1 ⇒ b1; w2 ⇒ b2; w3 ⇒ b3 }

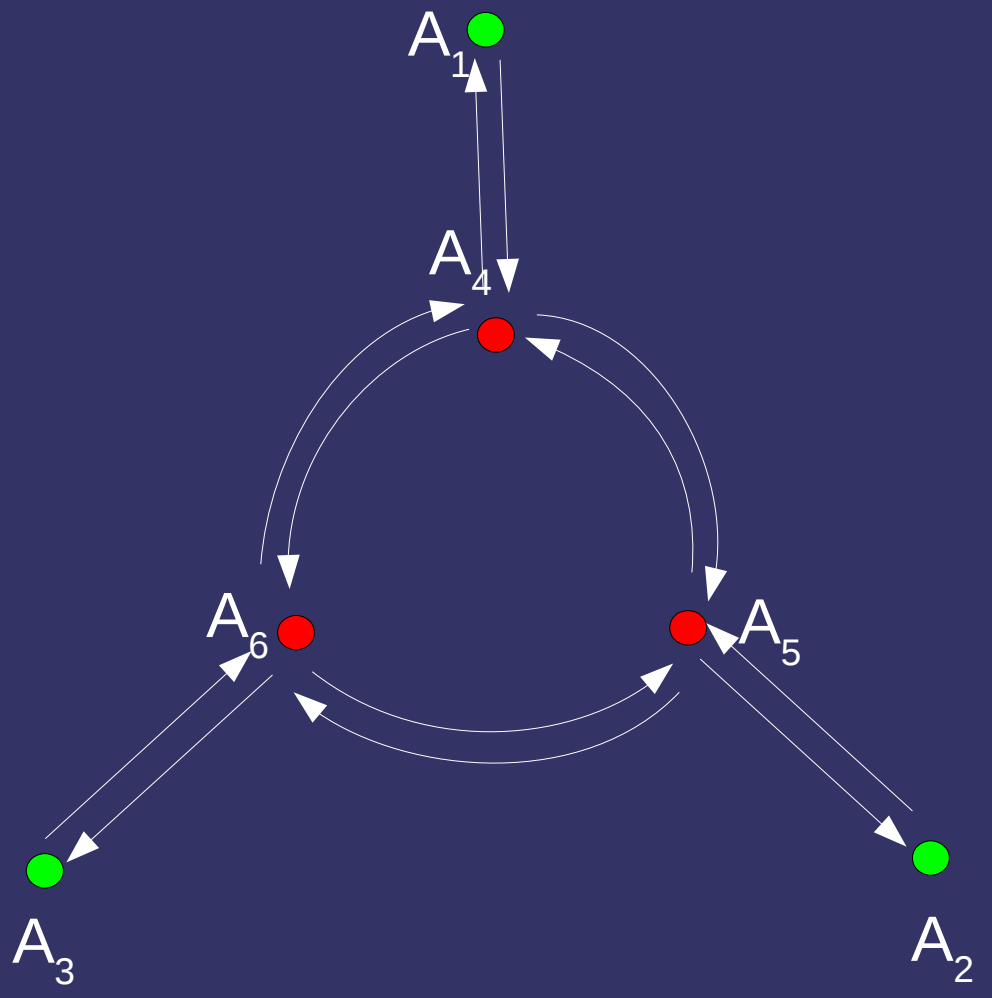
A_1: (w1) ⇒ b1
A_2: (w2) ⇒ b2
A_3: (w3) ⇒ b3
A_4: A_2, A_3 → ¬b1
A_5: A_1, A_3 → ¬b2
A_6: A_1, A_2 → ¬b3



Why Restricted Rebut

S = { w1; w2; w3; b2,b3→¬b1; b1,b3→¬b2; b1,b2→¬b3 }
D = { w1 ⇒ b1; w2 ⇒ b2; w3 ⇒ b3 }

A_1: (w1) ⇒ b1
A_2: (w2) ⇒ b2
A_3: (w3) ⇒ b3
A_4: A_2, A_3 → ¬b1
A_5: A_1, A_3 → ¬b2
A_6: A_1, A_2 → ¬b3

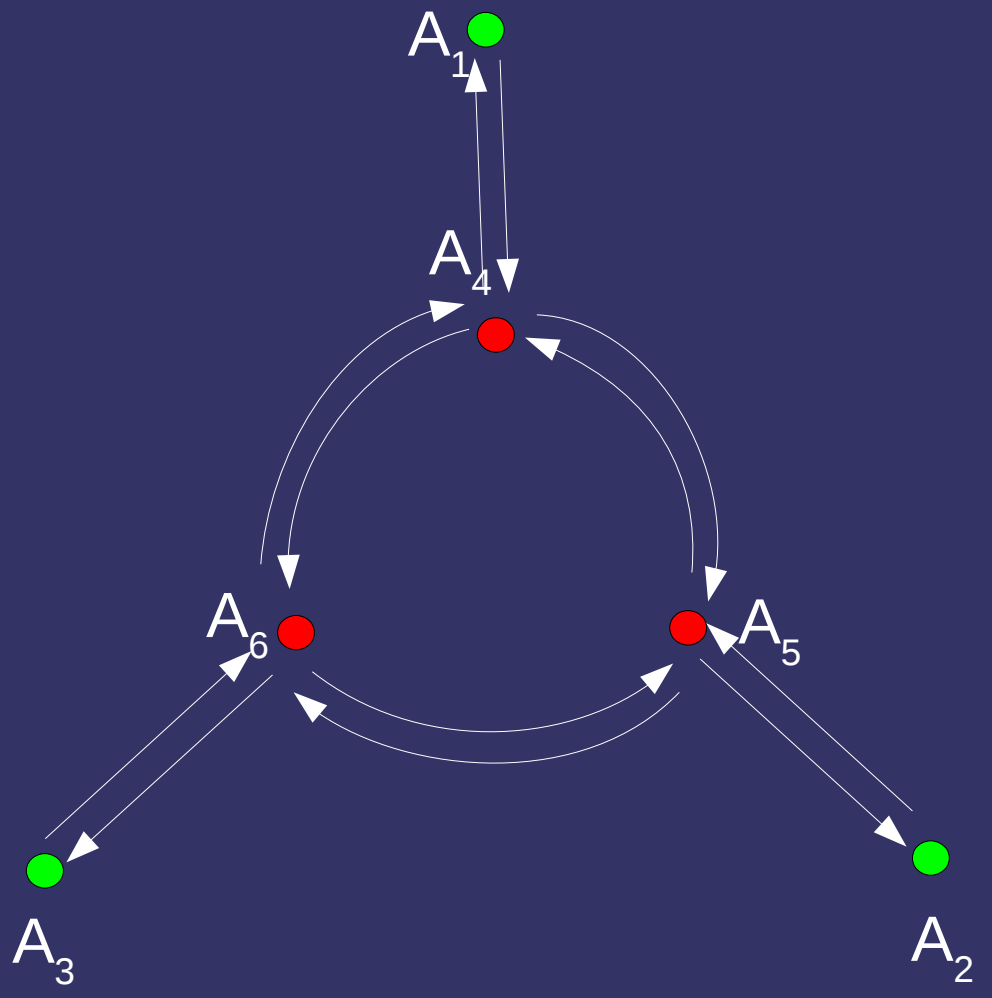


Why Restricted Rebut

S = { w1; w2; w3; b2,b3→¬b1; b1,b3→¬b2; b1,b2→¬b3 }
D = { w1 ⇒ b1; w2 ⇒ b2; w3 ⇒ b3 }

A_1: (w1) ⇒ b1
A_2: (w2) ⇒ b2
A_3: (w3) ⇒ b3
A_4: A_2, A_3 → ¬b1
A_5: A_1, A_3 → ¬b2
A_6: A_1, A_2 → ¬b3

b1 b2 b3 ¬b1 ¬b2 ¬b3



Why does this matter?

- Philosophy (as well as other science) aims to find statements that can be defended
- *“can be defended”* is usually left vague
- *“AI makes philosophy honest”* (Daniel Dennett)
- If you try to make things precise, all kinds of problems pop up…

What we found so far...

- when the aim is to apply unrestricted rebut and still have consistent outcome, only the *grounded* labelling will do (not preferred, stable, complete, ideal or CF2 labellings)
- a sound and complete discussion game for grounded labellings is available (details tomorrow)
- rule-strength ordering has to be either:
  - the empty ordering, or
  - a total ordering