An Introduction to Computational Argumentation Semantics (3/5)

Srdjan Vesic and Dragan Doder

ESSAI 2024

Attacks in practice

- When does an argument attack another one?
- Only two arguments: $\{a, b\}$

Attacks in practice

- When does an argument attack another one?
- Only two arguments: $\{a, b\}$
- No attacks: $\{a, b\}$ is the unique extension

Attacks in practice

- When does an argument attack another one?
- Only two arguments: $\{a, b\}$
- No attacks: $\{a, b\}$ is the unique extension
- If a attacks b but not vice versa: a in, b out

Attacks in practice

- When does an argument attack another one?
- Only two arguments: $\{a, b\}$
- No attacks: $\{a, b\}$ is the unique extension
- If a attacks b but not vice versa: a in, b out
- Bidirectional attack: two extensions

Attacks in practice

- Cramer and Guillaume
- Testing whether people agree
- Attack can be asymmetric
- Example: the source is not reliable

Find the attacks

- A: Louis applied the brake. Therefore, the car slowed down.
- B: Louis applied the accelerator instead of the brake. Therefore, Louis did not apply the brake.
- C: The car did not slow down. Therefore, the car approached the signal at the same speed or higher.
- D: Louis applied the clutch instead of the brake. Therefore, Louis did not apply the brake.

Find the attacks

In the following set of arguments, we assume that Marry has a pet named Maxy.

- A: Maxy's cage is glazed, so there is no hole in its cage.
- B: Maxy is a tiny snake, so Maxy can escape through the holes of its cage.
- C: The cage is specifically intended for tiny snakes; so all holes in the cage are too small for Maxy to go through.
- D: Marry put Maxy in a large cage, so Maxy cannot escape.

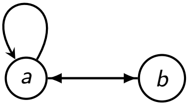
Find the attacks

L denotes the sentence “L is false”.

- A: If L were false, it would follow that “L is false” is true. But since L denotes the sentence “L is false”, this would mean that L is true. So if L were false, L would also be true, which is impossible. So L is not false. Therefore L is true.
- B: If L were true, this would mean that “L is false” is true, because L denotes the sentence “L is false”. But if “L is false” is true, it follows that L is false. So if L were true, L would also be false, which is impossible. So L is not true.

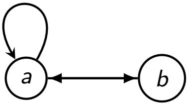
Ideal semantics

- A unique status semantics that returns more accepted arguments than grounded semantics?



Ideal semantics

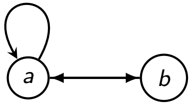
- A unique status semantics that returns more accepted arguments than grounded semantics?



- Ideal extension: a maximal admissible subset of every preferred extension

Ideal semantics

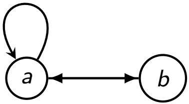
- A unique status semantics that returns more accepted arguments than grounded semantics?



- Ideal extension: a maximal admissible subset of every preferred extension
- Exercise: Find some ideal extensions

Ideal semantics

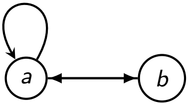
- A unique status semantics that returns more accepted arguments than grounded semantics?



- Ideal extension: a maximal admissible subset of every preferred extension
- Exercise: Find some ideal extensions
- Exercise: Prove that the ideal extension is unique

Ideal semantics

- A unique status semantics that returns more accepted arguments than grounded semantics?



- Ideal extension: a maximal admissible subset of every preferred extension
- Exercise: Find some ideal extensions
- Exercise: Prove that the ideal extension is unique
- Exercise: Find an example where the ideal extension is not equal to the intersection of all preferred extensions.

Semi-stable semantics

- Idea: We want stable extensions when they exists, and something similar when they don't

Semi-stable semantics

- Idea: We want stable extensions when they exists, and something similar when they don't
- For a set S, denote by $S^+ = \{x \mid S \text{ attacks } x\}$

Semi-stable semantics

- Idea: We want stable extensions when they exists, and something similar when they don't
- For a set S, denote by $S^+ = \{x \mid S \text{ attacks } x\}$
- S is a semi-stable extension if it is complete and $S \cup S^+$ is maximal for set inclusion among complete extensions, i.e. there is no complete extension S_1 such that $S \cup S^+ \subsetneq S_1 \cup S_1^+$

Semi-stable semantics

- Idea: We want stable extensions when they exists, and something similar when they don't
- For a set S, denote by $S^+ = \{x \mid S \text{ attacks } x\}$
- S is a semi-stable extension if it is complete and $S \cup S^+$ is maximal for set inclusion among complete extensions, i.e. there is no complete extension S_1 such that $S \cup S^+ \subsetneq S_1 \cup S_1^+$
- Exercise: Find some semi-stable extensions

Semi-stable semantics

- Idea: We want stable extensions when they exists, and something similar when they don't
- For a set S, denote by $S^+ = \{x \mid S \text{ attacks } x\}$
- S is a semi-stable extension if it is complete and $S \cup S^+$ is maximal for set inclusion among complete extensions, i.e. there is no complete extension S_1 such that $S \cup S^+ \subsetneq S_1 \cup S_1^+$
- Exercise: Find some semi-stable extensions
- Exercise: If there exists at least one stable extension, the sets of stable and semi-stable extensions coincide

Semi-stable semantics

- Idea: We want stable extensions when they exists, and something similar when they don't
- For a set S, denote by $S^+ = \{x \mid S \text{ attacks } x\}$
- S is a semi-stable extension if it is complete and $S \cup S^+$ is maximal for set inclusion among complete extensions, i.e. there is no complete extension S_1 such that $S \cup S^+ \subsetneq S_1 \cup S_1^+$
- Exercise: Find some semi-stable extensions
- Exercise: If there exists at least one stable extension, the sets of stable and semi-stable extensions coincide
- Exercise: If an argument a is in the grounded extension, it is in all semi-stable extensions

Labellings

Definition (Complete labelling)

We say that $\mathcal{L}ab$ is a complete labelling if and only if for every $a \in \mathcal{A}$:

- if a is labelled in then all its attackers are labelled out
- if a is labelled out then at least one of its attackers is labelled in
- if a is labelled undec then not all its attackers are labelled out and none of its attackers is labelled in.

We denote by $in(\mathcal{L}ab)$ (resp. $out(\mathcal{L}ab)$, $und(\mathcal{L}ab)$) the set of arguments labelled *in* (resp. *out*, *und*).

Exercise: labellings

Ext2Lab, Lab2Ext

- Ext2Lab: given an extension E, we define $\mathcal{L}ab = Ext2Lab(\mathcal{E})$ as follows:
  - $in(\mathcal{L}ab) = \mathcal{E}$
  - $out(\mathcal{L}ab) = \mathcal{E}^+$
  - $und(\mathcal{L}ab) = \mathcal{A} \setminus (\mathcal{E} \cup \mathcal{E}^+)$
- Lab2Ext: given a labelling $\mathcal{L}ab$, we define $\mathcal{E} = Lab2Ext(\mathcal{L}ab)$ as follows:
  - $\mathcal{E} = in(\mathcal{L}ab)$

Ext2Lab, Lab2Ext

- Ext2Lab: given an extension E, we define $\mathcal{L}ab = Ext2Lab(\mathcal{E})$ as follows:
  - $in(\mathcal{L}ab) = \mathcal{E}$
  - $out(\mathcal{L}ab) = \mathcal{E}^+$
  - $und(\mathcal{L}ab) = \mathcal{A} \setminus (\mathcal{E} \cup \mathcal{E}^+)$
- Lab2Ext: given a labelling $\mathcal{L}ab$, we define $\mathcal{E} = Lab2Ext(\mathcal{L}ab)$ as follows:
  - $\mathcal{E} = in(\mathcal{L}ab)$
- If $\mathcal{E}$ is a complete extension, $\texttt{Ext2}Lab(\mathcal{E})$ is a complete labelling

Ext2Lab, Lab2Ext

- Ext2Lab: given an extension E, we define $\mathcal{L}ab = Ext2Lab(\mathcal{E})$ as follows:
  - $in(\mathcal{L}ab) = \mathcal{E}$
  - $out(\mathcal{L}ab) = \mathcal{E}^+$
  - $und(\mathcal{L}ab) = \mathcal{A} \setminus (\mathcal{E} \cup \mathcal{E}^+)$
- Lab2Ext: given a labelling $\mathcal{L}ab$, we define $\mathcal{E} = Lab2Ext(\mathcal{L}ab)$ as follows:
  - $\mathcal{E} = in(\mathcal{L}ab)$
- If $\mathcal{E}$ is a complete extension, $\texttt{Ext2}Lab(\mathcal{E})$ is a complete labelling
- If $\mathcal{L}ab$ is a complete labelling, $\mathcal{L}ab2Ext(\mathcal{L}ab)$ is a complete extension

Other semantics in labelling approach

- The grounded labelling is a complete labelling $\mathcal{L}ab$ where $in(\mathcal{L}ab)$ is minimal for set inclusion

Other semantics in labelling approach

- The grounded labelling is a complete labelling $\mathcal{L}ab$ where $in(\mathcal{L}ab)$ is minimal for set inclusion
- A preferred labelling is a complete labelling such that the set of arguments labelled *in* is maximal with respect to set inclusion among all complete labellings

Other semantics in labelling approach

- The grounded labelling is a complete labelling $\mathcal{L}ab$ where $in(\mathcal{L}ab)$ is minimal for set inclusion
- A preferred labelling is a complete labelling such that the set of arguments labelled *in* is maximal with respect to set inclusion among all complete labellings
- A stable labelling is a complete labelling such that

Other semantics in labelling approach

- The grounded labelling is a complete labelling $\mathcal{L}ab$ where $in(\mathcal{L}ab)$ is minimal for set inclusion
- A preferred labelling is a complete labelling such that the set of arguments labelled *in* is maximal with respect to set inclusion among all complete labellings
- A stable labelling is a complete labelling such that the set of undecided arguments is empty

Other semantics in labelling approach

- The grounded labelling is a complete labelling $\mathcal{L}ab$ where $in(\mathcal{L}ab)$ is minimal for set inclusion
- A preferred labelling is a complete labelling such that the set of arguments labelled *in* is maximal with respect to set inclusion among all complete labellings
- A stable labelling is a complete labelling such that the set of undecided arguments is empty
- A semi-stable labelling is a complete labelling such that

Other semantics in labelling approach

- The grounded labelling is a complete labelling $\mathcal{L}ab$ where $in(\mathcal{L}ab)$ is minimal for set inclusion
- A preferred labelling is a complete labelling such that the set of arguments labelled *in* is maximal with respect to set inclusion among all complete labellings
- A stable labelling is a complete labelling such that the set of undecided arguments is empty
- A semi-stable labelling is a complete labelling such that undec is minimal