A dynamic approach for combining abstract argumentation semantics

Second Chinese Conference on Logic and Argumentation, Hangzhou, 2018

Jérémie Dauphin, Marcos Cramer and Leon van der Torre

University of Luxembourg

June 16, 2018

Motivations

- Increasing number of semantics: 16 different semantics described in the Handbook of Formal Argumentation.
- Diversity is good, but how to choose?
- Principle-based approaches help, but sometimes no semantics fits perfectly.

Motivations

- Increasing number of semantics: 16 different semantics described in the Handbook of Formal Argumentation.
- Diversity is good, but how to choose?
- Principle-based approaches help, but sometimes no semantics fits perfectly.

We propose a formal framework for dynamically combining argumentation semantics.

Update semantics

Direct Semantics

From unlabelled to fully labelled frameworks in a single step.

Update semantics

We decompose this relation into a more fine-grained one, while respecting the principles of:

- Monotonic precision increase
- Reachable fixpoints are fully labelled

$$LAF_1 \longrightarrow LAF_2 \longrightarrow \cdots \longrightarrow LAF_f \circlearrowleft$$

Update semantics

Direct Semantics

From unlabelled to fully labelled frameworks in a single step.

Update semantics

We decompose this relation into a more fine-grained one, while respecting the principles of:

- Monotonic precision increase
- Reachable fixpoints are fully labelled

$$LAF_1 \longrightarrow LAF_2 \longrightarrow \cdots \longrightarrow LAF_f \circlearrowleft$$

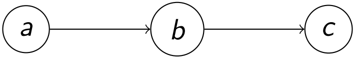
We say that an update semantics *gives rise to* a direct semantics iff the reachable fixpoints correspond exactly to the direct labellings.

Preliminaries

Abstract Argumentation Framework

A pair (A, R):

- A: a set of *abstract* arguments;
- $R \subseteq A \times A$: a relation of *attack* between arguments.



Labellings

Labels

Assign a single label to each argument:

- in for accepted
- out for rejected
- undec for undecided

Semantics

Admissibility

A set of arguments S is admissible iff:

- it defends all its arguments;
- it is conflict-free.

Semantics

Admissibility

A set of arguments S is admissible iff:

- it defends all its arguments;
- it is conflict-free.

Labelling-based abstract argumentation semantics

Attach a label to every argument

- Complete: admissible + contains all arguments it defends
- Grounded: Complete + minimal *in*
- Preferred: Complete + maximal *in*

A concrete update semantics

The step_grounded semantics

Procedure:

1. Look for unlabelled arguments with all attackers out, make them in and anything they attack out;
2. Otherwise, label all remaining arguments undec.

A concrete update semantics

The step_grounded semantics

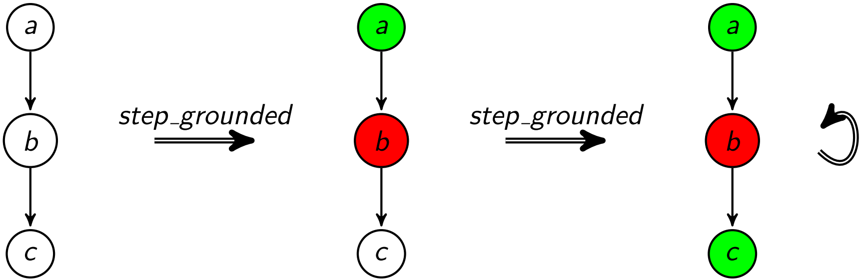
Procedure:

1. Look for unlabelled arguments with all attackers out, make them in and anything they attack out;
2. Otherwise, label all remaining arguments undec.

Theorem 1

step_grounded is an update semantics which gives rise to the grounded semantics.

Example



Towards a step_preferred semantics

Minimal admissible sets

- Admissible;
- Non-empty;
- Unlabelled;
- Minimally so.

Second update semantics

The step_preferred semantics

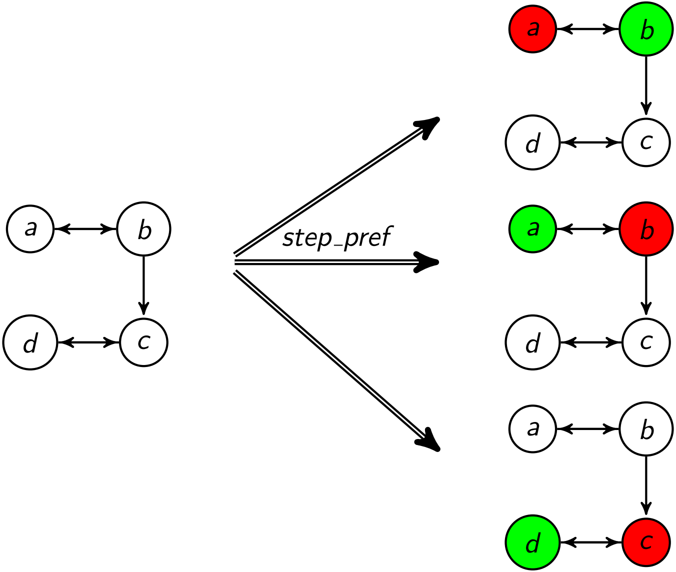
Procedure:

1. Look for unlabelled arguments with all attackers out, make them in and anything they attack out;
2. Else, look for minimal admissible sets of arguments, label those in and anything they attack out;
3. Otherwise, label all remaining arguments undec.

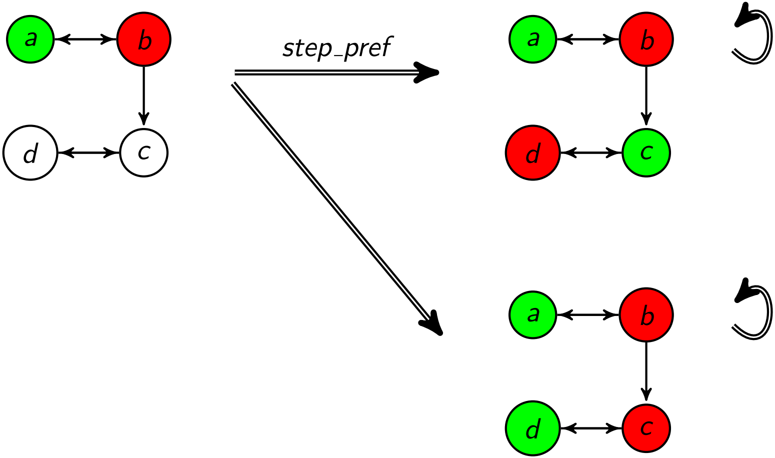
Theorem 2

step_preferred is an update semantics which gives rise to the preferred semantics.

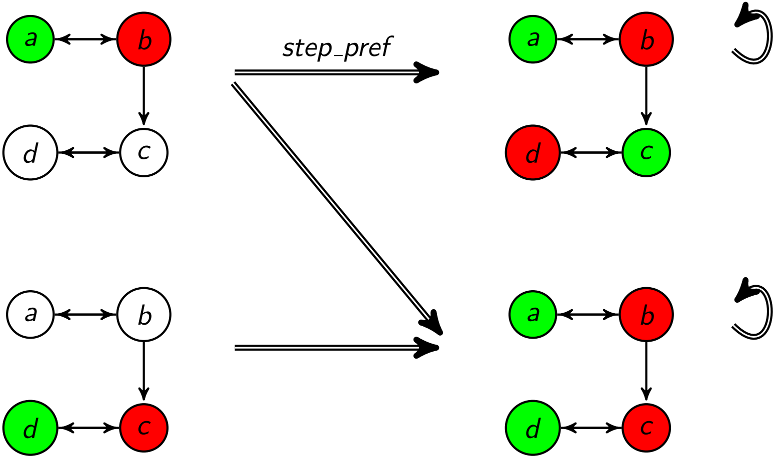
Example



Example



Example

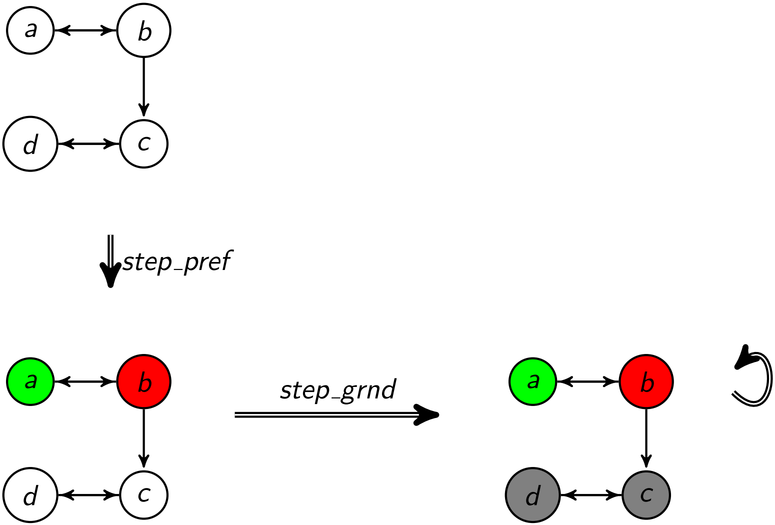


Composition of update semantics

Theorem 3

step_grounded ∪ *step_preferred* is an update semantics which gives rise to the complete semantics.

Example



A semantic approach

Precision ordering over the partially-labelled frameworks

$(\langle A, R\rangle, Lab) \geq_p (\langle A', R'\rangle, Lab')$ iff:

- $\langle A, R\rangle = \langle A', R'\rangle$;
- $\forall a \in A$, if $Lab(a')$ is defined, then $Lab(a) = Lab(a')$.

A semantic approach

Precision ordering over the partially-labelled frameworks

$(\langle A, R\rangle, Lab) \geq_p (\langle A', R'\rangle, Lab')$ iff:

- $\langle A, R\rangle = \langle A', R'\rangle$;
- $\forall a \in A$, if $Lab(a')$ is defined, then $Lab(a) = Lab(a')$.

The most fine-grained update semantics

For a given direct semantics *dir*, we define mfg_{dir} as the transitive reduction of $\geq_p$ restricted to the paths from initial frameworks to the corresponding final ones returned by *dir*.

Combining most fine-grained updates

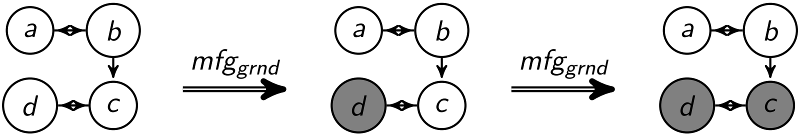
Combination operation

Given u_1 and u_2, we define $u_1 \uplus u_2$.
We also allow for changes to be imported into a framework F from a similar framework F_2, if the framework can be partitioned into $S, I, M \subseteq A$:

1. I is already fully labelled;
2. there is no conflict between S and M;
3. F and F_2 agree on $S \cup I$;
4. the update happens in S;
5. the current label of $I \cup M$ is reachable by $u_1 \uplus u_2$.

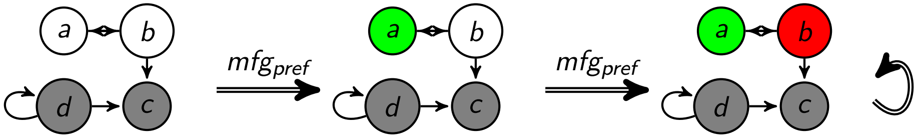
Example

F:



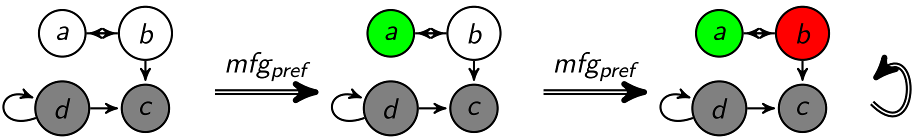
Example continued

F_2:

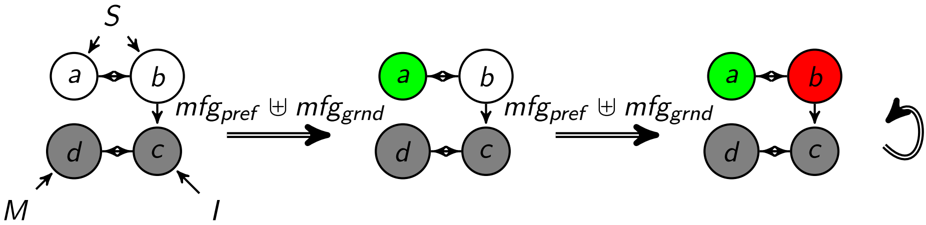


Example continued

F_2:



F:



Conclusions and future work

Conclusions:

- Decomposed some direct semantics into fine-grained version.
- Combined preferred and grounded to get complete in two different ways. What does this mean for direct semantics?

Future work:

- Define update semantics based on algorithms, such as SCC-recursiveness.
- Examine the results of other combinations of update semantics.
- How fine-grained do you need to be in order to get interesting combinations of updates?
- A model closer to learning-oriented reasoning?

Conclusions and future work

Conclusions:

- Decomposed some direct semantics into fine-grained version.
- Combined preferred and grounded to get complete in two different ways. What does this mean for direct semantics?

Future work:

- Define update semantics based on algorithms, such as SCC-recursiveness.
- Examine the results of other combinations of update semantics.
- How fine-grained do you need to be in order to get interesting combinations of updates?
- A model closer to learning-oriented reasoning?

Thank you!