Incremental Computation of Deterministic Extensions for Dynamic Argumentation Frameworks

Sergio Greco and Francesco Parisi

email:{greco, fparisi}@dimes.unical.it
Department of Informatics, Modeling, Electronics and System Engineering
University of Calabria
Italy

15^{th} European Conference On Logics In Artificial Intelligence

November 9-11, 2016

Larnaca, Cyprus

Argumentation in AI

- A general way for representing arguments and relationships (rebuttals) between them
- It allows representing dialogues, making decisions, and handling inconsistency and uncertainty

Abstract Argumentation Framework (AF) [Dung 1995]: arguments are abstract entities (no attention is paid to their internal structure) that may attack and/or be attacked by other arguments

Example (a simple AF)

a = Our friends will have great fun at our party on Saturday
b = Saturday will rain (according to the weather forecasting service 1)
c = Saturday will be sunny (according to the weather forecasting service 2)

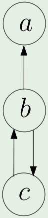
Argumentation in AI

- A general way for representing arguments and relationships (rebuttals) between them
- It allows representing dialogues, making decisions, and handling inconsistency and uncertainty

Abstract Argumentation Framework (AF) [Dung 1995]: arguments are abstract entities (no attention is paid to their internal structure) that may attack and/or be attacked by other arguments

Example (a simple AF)

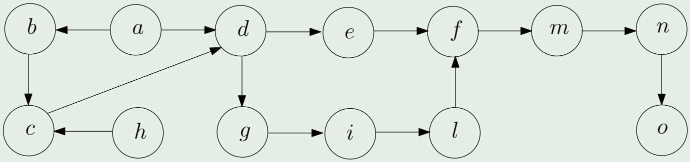
a = Our friends will have great fun at our party on Saturday
b = Saturday will rain (according to the weather forecasting service 1)
c = Saturday will be sunny (according to the weather forecasting service 2)



Semantics for Abstract Argumentations

- Several semantics have been proposed to identify "reasonable" sets of arguments (called *extensions*)

Example (AF $\mathcal{A}_0$)



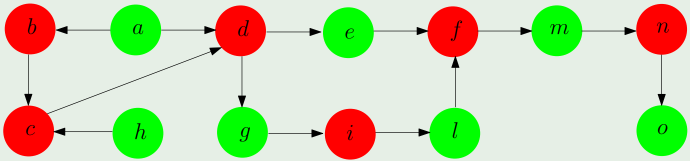
$E_0 = \{a, h, g, e, l, m, o\}$ is an extension according to the most popular semantics, i.e. *grounded*, *complete*, *ideal*, *preferred*, *stable*, and *semi-stable*

- Extensions change if we update the initial AF by adding/removing arguments/attacks
- **Should we recompute the semantics of updated AFs from scratch?**

Semantics for Abstract Argumentations

- Several semantics have been proposed to identify "reasonable" sets of arguments (called *extensions*)

Example (AF $\mathcal{A}_0$)



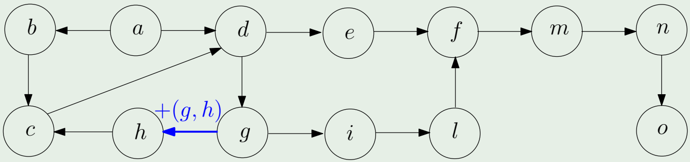
$E_0 = \{a, h, g, e, l, m, o\}$ is an extension according to the most popular semantics, i.e. *grounded*, *complete*, *ideal*, *preferred*, *stable*, and *semi-stable*

- Extensions change if we update the initial AF by adding/removing arguments/attacks
- **Should we recompute the semantics of updated AFs from scratch?**

Semantics for Abstract Argumentations

- Several semantics have been proposed to identify “reasonable” sets of arguments (called *extensions*)

Example (Updated AF $\mathcal{A} = +(g, h)(\mathcal{A}_0)$)



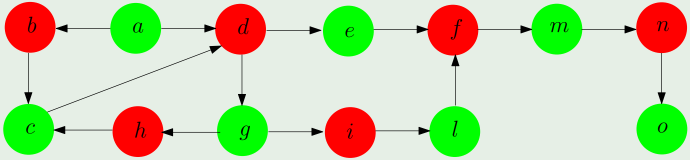
We need to recompute the extension E of the AF obtained by adding attack (g, h)

- Extensions change if we update the initial AF by adding/removing arguments/attacks
- Should we recompute the semantics of updated AFs from scratch?

Semantics for Abstract Argumentations

- Several semantics have been proposed to identify “reasonable” sets of arguments (called *extensions*)

Example (Updated AF $\mathcal{A} = +(g, h)(\mathcal{A}_0)$)



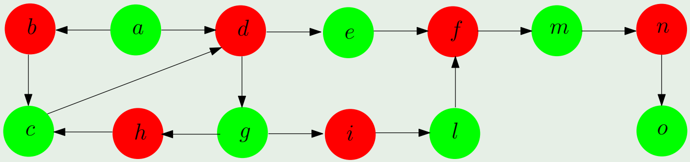
$E = \{a, c, g, e, l, m, o\}$ is an extension according to the most popular semantics, i.e. *grounded*, *complete*, *ideal*, *preferred*, *stable*, and *semi-stable*

- Extensions change if we update the initial AF by adding/removing arguments/attacks
- Should we recompute the semantics of updated AFs from scratch?

Semantics for Abstract Argumentations

- Several semantics have been proposed to identify “reasonable” sets of arguments (called *extensions*)

Example (Updated AF $\mathcal{A} = +(g, h)(\mathcal{A}_0)$)



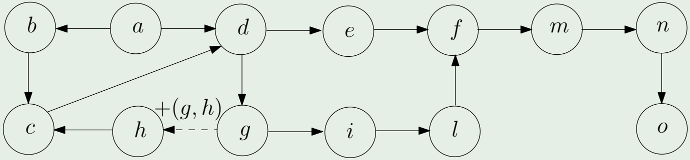
$E = \{a, c, g, e, l, m, o\}$ is an extension according to the most popular semantics, i.e. *grounded*, *complete*, *ideal*, *preferred*, *stable*, and *semi-stable*

- Extensions change if we update the initial AF by adding/removing arguments/attacks
- **Should we recompute the semantics of updated AFs from scratch?**

Influenced set

- For the grounded and ideal semantics, the extension E can be efficiently computed incrementally by looking only at a small part of the AF, which is "influenced by" the update operation.

Example (AF $\mathcal{A}_0$ and updated AF $\mathcal{A} = +(g, h)(\mathcal{A}_0)$)



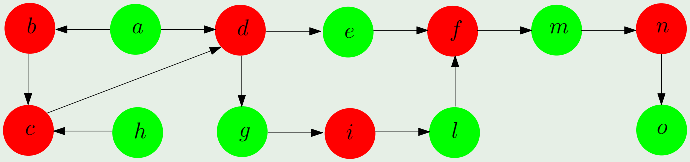
The influenced set is just $\{h, c\}$: only the status of h and c changes

- The influenced set refines the previously proposed set of *affected arguments* [Liao et al. 2011, Baroni et. al. 2014]
- In the example, all the arguments but a and b turn out to be affected

Influenced set

- For the grounded and ideal semantics, the extension E can be efficiently computed incrementally by looking only at a small part of the AF, which is "influenced by" the update operation.

Example (AF $\mathcal{A}_0$ and updated AF $\mathcal{A} = +(g, h)(\mathcal{A}_0)$)



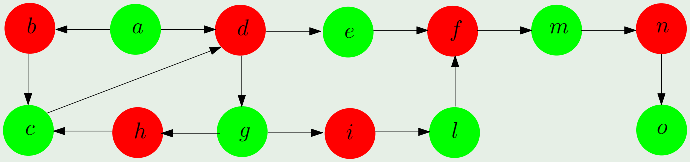
The influenced set is just $\{h, c\}$: only the status of h and c changes

- The influenced set refines the previously proposed set of *affected arguments* [Liao et al. 2011, Baroni et. al. 2014]
- In the example, all the arguments but a and b turn out to be affected

Influenced set

- For the grounded and ideal semantics, the extension E can be efficiently computed incrementally by looking only at a small part of the AF, which is "influenced by" the update operation.

Example (AF $\mathcal{A}_0$ and updated AF $\mathcal{A} = +(g, h)(\mathcal{A}_0)$)



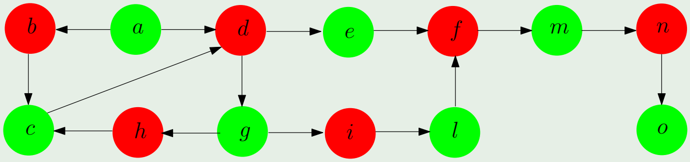
The influenced set is just $\{h, c\}$: only the status of h and c changes

- The influenced set refines the previously proposed set of *affected arguments* [Liao et al. 2011, Baroni et. al. 2014]
- In the example, all the arguments but a and b turn out to be affected

Influenced set

- For the grounded and ideal semantics, the extension *E* can be efficiently computed incrementally by looking only at a small part of the AF, which is "influenced by" the update operation.

Example (AF $\mathcal{A}_0$ and updated AF $\mathcal{A} = +(g, h)(\mathcal{A}_0)$)



The influenced set is just $\{h, c\}$: only the status of *h* and *c* changes

- The influenced set refines the previously proposed set of *affected arguments* [Liao et al. 2011, Baroni et. al. 2014]
- In the example, all the arguments but *a* and *b* turn out to be affected

Incremental algorithms and experiments

- We formally define the concept of *influenced set* consisting of the arguments whose status could change after an update.
- We focus on the grounded and ideal semantics, which are deterministic (admit exactly one extension)
- We present an incremental algorithm for recomputing the grounded extension; it computes the status of influenced arguments only.
- We present an incremental algorithm for the efficient recomputation of the ideal semantics which takes advantage of both the set of influenced arguments and the efficient algorithm for computing grounded extensions.
- Experimental results show the effectiveness of our approach.

Outline

1 Introduction
- Motivation
- Contributions

2 Preliminaries
- Abstract Argumentation Frameworks
- Updates

3 Algorithms
- Influenced Arguments
- Incremental Computation of Grounded Semantics
- Incremental Computation of Ideal Semantics

4 Experiments

5 Conclusions and future work
- References

Basic concepts

- An *(abstract) argumentation framework* (*AF*) is a pair $\langle A, \Sigma \rangle$, where A is a set of *arguments* and $\Sigma \subseteq A \times A$ is a set of *attacks*.

Example (AAF)

$A = \{a, b, c, d\}$
$\Sigma = \{(a, b), (b, a), (b, c), (d, c)\}$

- A set $S \subseteq A$ is *conflict-free* if there are no $a, b \in S$ such that *a attacks b*
- *S defends a* iff $\forall b \in A$ that *attacks a* there is $c \in S$ that *attacks b*
- S is *admissible* if it is conflict-free and it defends all its arguments.

Example (conflict-free and admissible sets)

- $\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, c\}, \{a, d\}, \{b, d\}$ are conflict-free
- $\{a\}$ defend a; $\{b, d\}$ defends both b and d
- $\emptyset, \{a\}, \{d\}, \{a, d\}, \{b, d\}$ are admissible

Basic concepts

- An *(abstract) argumentation framework* (*AF*) is a pair $\langle A, \Sigma \rangle$, where A is a set of *arguments* and $\Sigma \subseteq A \times A$ is a set of *attacks*.

Example (AAF)

$A = \{a, b, c, d\}$
$\Sigma = \{(a, b), (b, a), (b, c), (d, c)\}$

- A set $S \subseteq A$ is *conflict-free* if there are no $a, b \in S$ such that *a attacks b*
- *S defends a* iff $\forall b \in A$ that *attacks a* there is $c \in S$ that *attacks b*
- *S* is *admissible* if it is conflict-free and it defends all its arguments.

Example (conflict-free and admissible sets)

- $\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, c\}, \{a, d\}, \{b, d\}$ are conflict-free
- $\{a\}$ defend a; $\{b, d\}$ defends both b and d
- $\emptyset, \{a\}, \{d\}, \{a, d\}, \{b, d\}$ are admissible

Basic concepts

- An *(abstract) argumentation framework* (*AF*) is a pair $\langle A, \Sigma \rangle$, where A is a set of *arguments* and $\Sigma \subseteq A \times A$ is a set of *attacks*.

Example (AAF)

$A = \{a, b, c, d\}$
$\Sigma = \{(a, b), (b, a), (b, c), (d, c)\}$

- A set $S \subseteq A$ is *conflict-free* if there are no $a, b \in S$ such that *a attacks b*
- *S defends a* iff $\forall b \in A$ that *attacks a* there is $c \in S$ that *attacks b*
- S is *admissible* if it is conflict-free and it defends all its arguments.

Example (conflict-free and admissible sets)

- $\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, c\}, \{a, d\}, \{b, d\}$ are conflict-free
- $\{a\}$ defend a; $\{b, d\}$ defends both b and d
- $\emptyset, \{a\}, \{d\}, \{a, d\}, \{b, d\}$ are admissible

Basic concepts

- An *(abstract) argumentation framework* (*AF*) is a pair $\langle A, \Sigma \rangle$, where A is a set of *arguments* and $\Sigma \subseteq A \times A$ is a set of *attacks*.

Example (AAF)

$A = \{a, b, c, d\}$
$\Sigma = \{(a, b), (b, a), (b, c), (d, c)\}$

- A set $S \subseteq A$ is *conflict-free* if there are no $a, b \in S$ such that *a attacks b*
- *S defends a* iff $\forall b \in A$ that *attacks a* there is $c \in S$ that *attacks b*
- *S* is *admissible* if it is conflict-free and it defends all its arguments.

Example (conflict-free and admissible sets)

- $\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, c\}, \{a, d\}, \{b, d\}$ are conflict-free
- $\{a\}$ defend a; $\{b, d\}$ defends both b and d
- $\emptyset, \{a\}, \{d\}, \{a, d\}, \{b, d\}$ are admissible

Semantics for Abstract Argumentation

A semantics identifies "reasonable" sets of arguments, called *extensions*

- A *complete extension* ($\mathtt{co}$) is an admissible set that contains all the arguments that it defends.

A complete extension S is said to be:

- *preferred* ($\mathtt{pr}$) iff it is maximal
- *semi-stable* ($\mathtt{ss}$) iff $S \cup S^+$ is maximal (S^+ are arguments attacked by S)
- *stable* ($\mathtt{st}$) iff it attacks each argument in $A \setminus S$
- *grounded* ($\mathtt{gr}$) iff it is minimal
- *ideal* ($\mathtt{id}$) iff it is contained in every preferred extension and it is maximal

Example (semantics for AAF)

complete extensions: $\{d\}, \{a, d\}, \{b, d\}$

preferred extensions: $\{a, d\}, \{b, d\}$
semi-stable extensions: $\{a, d\}, \{b, d\}$
stable extensions: $\{a, d\}, \{b, d\}$
grounded extension: $\{d\}$
ideal extension: $\{d\}$

Semantics for Abstract Argumentation

A semantics identifies “reasonable” sets of arguments, called *extensions*

- A *complete extension* ($\texttt{co}$) is an admissible set that contains all the arguments that it defends.

A complete extension S is said to be:

- *preferred ($\texttt{pr}$)* iff it is maximal
- *semi-stable ($\texttt{ss}$)* iff $S \cup S^+$ is maximal (S^+ are arguments attacked by S)
- *stable ($\texttt{st}$)* iff it attacks each argument in $A \setminus S$
- *grounded ($\texttt{gr}$)* iff it is minimal
- *ideal ($\texttt{id}$)* iff it is contained in every preferred extension and it is maximal

Example (semantics for AAF)

complete extensions: $\{d\}, \{a, d\}, \{b, d\}$
preferred extensions: $\{a, d\}, \{b, d\}$
semi-stable extensions: $\{a, d\}, \{b, d\}$
stable extensions: $\{a, d\}, \{b, d\}$

grounded extension: $\{d\}$
ideal extension: $\{d\}$

Semantics for Abstract Argumentation

A semantics identifies “reasonable” sets of arguments, called *extensions*

- A *complete extension* ($\mathtt{co}$) is an admissible set that contains all the arguments that it defends.

A complete extension S is said to be:

- *preferred* ($\mathtt{pr}$) iff it is maximal
- *semi-stable* ($\mathtt{ss}$) iff $S \cup S^+$ is maximal (S^+ are arguments attacked by S)
- *stable* ($\mathtt{st}$) iff it attacks each argument in $A \setminus S$
- *grounded* ($\mathtt{gr}$) iff it is minimal
- *ideal* ($\mathtt{id}$) iff it is contained in every preferred extension and it is maximal

Example (semantics for AAF)

complete extensions: $\{d\}, \{a, d\}, \{b, d\}$
preferred extensions: $\{a, d\}, \{b, d\}$
semi-stable extensions: $\{a, d\}, \{b, d\}$
stable extensions: $\{a, d\}, \{b, d\}$
grounded extension: $\{d\}$
ideal extension: $\{d\}$

Semantics for Abstract Argumentation

A semantics identifies “reasonable” sets of arguments, called *extensions*

- A *complete extension* (co) is an admissible set that contains all the arguments that it defends.

A complete extension S is said to be:

- *preferred (pr)* iff it is maximal
- *semi-stable (ss)* iff $S \cup S^+$ is maximal (S^+ are arguments attacked by S)
- *stable (st)* iff it attacks each argument in $A \setminus S$
- *grounded (gr)* iff it is minimal
- *ideal (id)* iff it is contained in every preferred extension and it is maximal

Example (semantics for AAF)

complete extensions: $\{d\}, \{a, d\}, \{b, d\}$
preferred extensions: $\{a, d\}, \{b, d\}$
semi-stable extensions: $\{a, d\}, \{b, d\}$
stable extensions: $\{a, d\}, \{b, d\}$

grounded extension: $\{d\}$
ideal extension: $\{d\}$

Semantics for Abstract Argumentation

A semantics identifies “reasonable” sets of arguments, called *extensions*

- A *complete extension* ($\mathtt{co}$) is an admissible set that contains all the arguments that it defends.

A complete extension S is said to be:

- *preferred* ($\mathtt{pr}$) iff it is maximal
- *semi-stable* ($\mathtt{ss}$) iff $S \cup S^+$ is maximal (S^+ are arguments attacked by S)
- *stable* ($\mathtt{st}$) iff it attacks each argument in $A \setminus S$
- *grounded* ($\mathtt{gr}$) iff it is minimal
- *ideal* ($\mathtt{id}$) iff it is contained in every preferred extension and it is maximal

Example (semantics for AAF)

complete extensions: $\{d\}, \{a, d\}, \{b, d\}$
preferred extensions: $\{a, d\}, \{b, d\}$
semi-stable extensions: $\{a, d\}, \{b, d\}$
stable extensions: $\{a, d\}, \{b, d\}$

grounded extension: $\{d\}$
ideal extension: $\{d\}$

Semantics for Abstract Argumentation

A semantics identifies "reasonable" sets of arguments, called *extensions*

- A *complete extension* ($\mathtt{co}$) is an admissible set that contains all the arguments that it defends.

A complete extension S is said to be:

- *preferred* ($\mathtt{pr}$) iff it is maximal
- *semi-stable* ($\mathtt{ss}$) iff $S \cup S^+$ is maximal (S^+ are arguments attacked by S)
- *stable* ($\mathtt{st}$) iff it attacks each argument in $A \setminus S$
- *grounded* ($\mathtt{gr}$) iff it is minimal
- *ideal* ($\mathtt{id}$) iff it is contained in every preferred extension and it is maximal

Example (semantics for AAF)

complete extensions: $\{d\}, \{a, d\}, \{b, d\}$
preferred extensions: $\{a, d\}, \{b, d\}$
semi-stable extensions: $\{a, d\}, \{b, d\}$
stable extensions: $\{a, d\}, \{b, d\}$
grounded extension: $\{d\}$
ideal extension: $\{d\}$

Deterministic (or unique status) semantics

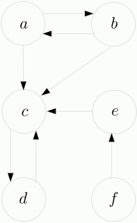
- All the semantics except the stable admit at least one extension
- Grounded and ideal semantics admit exactly one extension
- Semantics `gr` and `id` are called *deterministic* or *unique status*

Example (Multiple status vs unique status semantics)

3 complete extensions: $\{d\}, \{a, d\}, \{b, d\}$
2 preferred extensions: $\{a, d\}, \{b, d\}$
2 semi-stable extensions: $\{a, d\}, \{b, d\}$
2 stable extensions: $\{a, d\}, \{b, d\}$

1 grounded extension: $\{d\}$
1 ideal extension: $\{d\}$

Example (Deterministic semantics)



grounded extension: $\{f\}$

ideal extension: $\{d, f\}$

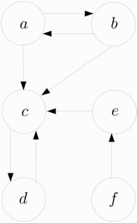
Deterministic (or unique status) semantics

- All the semantics except the stable admit at least one extension
- Grounded and ideal semantics admit exactly one extension
- Semantics `gr` and `id` are called *deterministic* or *unique status*

Example (Multiple status vs unique status semantics)

3 complete extensions: $\{d\}, \{a, d\}, \{b, d\}$
2 preferred extensions: $\{a, d\}, \{b, d\}$
2 semi-stable extensions: $\{a, d\}, \{b, d\}$
2 stable extensions: $\{a, d\}, \{b, d\}$
1 grounded extension: $\{d\}$
1 ideal extension: $\{d\}$

Example (Deterministic semantics)



grounded extension: $\{f\}$

ideal extension: $\{d, f\}$

Deterministic (or unique status) semantics

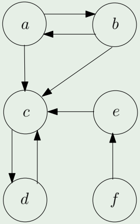
- All the semantics except the stable admit at least one extension
- Grounded and ideal semantics admit exactly one extension
- Semantics `gr` and `id` are called *deterministic* or *unique status*

Example (Multiple status vs unique status semantics)

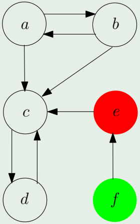
3 complete extensions: $\{d\}, \{a, d\}, \{b, d\}$
2 preferred extensions: $\{a, d\}, \{b, d\}$
2 semi-stable extensions: $\{a, d\}, \{b, d\}$
2 stable extensions: $\{a, d\}, \{b, d\}$

1 grounded extension: $\{d\}$
1 ideal extension: $\{d\}$

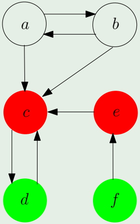
Example (Deterministic semantics)



grounded extension: $\{f\}$



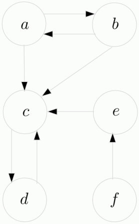
ideal extension: $\{d, f\}$



Extensions and labellings

- Semantics can be also defined in terms of *labelling*.
- Function $L : A \to \{\text{IN}, \text{OUT}, \text{UN}\}$ assigns a label to each argument
  - $L(a) = \text{IN}$ means a is accepted (i.e., *all* arguments attacking a are rejected)
  - $L(a) = \text{OUT}$ means a is rejected (i.e., an argument attacking a is accepted)
  - $L(a) = \text{UN}$ means that a is undecided
- Extension E corresponds to the labelling $L = \langle E, E^+, A \setminus (E \cup E^+) \rangle$
- Labelling L corresponds to the extension consisting of the arguments labelled as IN

Example (Two complete labellings: the grounded and the ideal labelling)



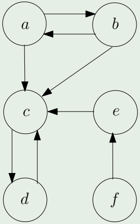
grounded extension: $\{f\}$

ideal extension: $\{d, f\}$

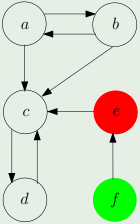
Extensions and labellings

- Semantics can be also defined in terms of *labelling*.
- Function $L : A \to \{\text{IN}, \text{OUT}, \text{UN}\}$ assigns a label to each argument
  - $L(a) = \text{IN}$ means a is accepted (i.e., *all* arguments attacking a are rejected)
  - $L(a) = \text{OUT}$ means a is rejected (i.e., an argument attacking a is accepted)
  - $L(a) = \text{UN}$ means that a is undecided
- Extension E corresponds to the labelling $L = \langle E, E^+, A \setminus (E \cup E^+)\rangle$
- Labelling L corresponds to the extension consisting of the arguments labelled as IN

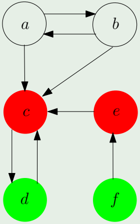
Example (Two complete labellings: the grounded and the ideal labelling)



grounded extension: $\{f\}$



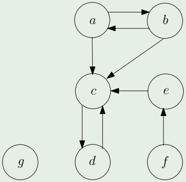
ideal extension: $\{d, f\}$



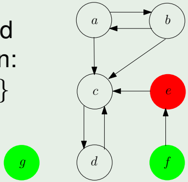
Updates

- An *update u* for an AF $\mathcal{A}_0$ consists in modifying $\mathcal{A}_0$ into an AF $\mathcal{A}$ by adding or removing arguments or attacks.
- If E_0 is an extension for $\mathcal{A}_0$ and $\mathcal{A}$ is obtained by adding (resp. removing) the set S of isolated arguments, then $E = E_0 \cup S$ (resp. $E = E_0 \setminus S$)
- We focus on the addition $+(a, b)$ and deletion $-(a, b)$ of an attack (a, b)
- $u(\mathcal{A}_0)$ denotes the application of update $u = \pm(a, b)$ to $\mathcal{A}_0$.
- Multiple updates $\{+(a_1, b_1), \ldots, +(a_n, b_n), -(a'_1, b'_1), \ldots, -(a'_m, b'_m)\}$ can be simulated by a single attack update

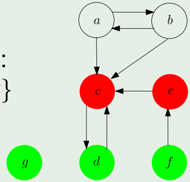
Example (Extensions/labellings after adding the isolated argument g)



grounded extension: $\{f\} \cup \{g\}$



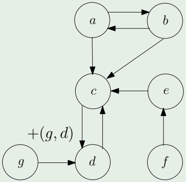
ideal extension: $\{d, f\} \cup \{g\}$



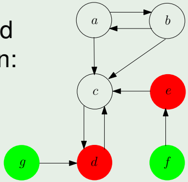
Updates

- An *update* u for an AF $\mathcal{A}_0$ consists in modifying $\mathcal{A}_0$ into an AF $\mathcal{A}$ by adding or removing arguments or attacks.
- If E_0 is an extension for $\mathcal{A}_0$ and $\mathcal{A}$ is obtained by adding (resp. removing) the set S of isolated arguments, then $E = E_0 \cup S$ (resp. $E = E_0 \setminus S$)
- We focus on the addition $+(a, b)$ and deletion $-(a, b)$ of an attack (a, b)
- $u(\mathcal{A}_0)$ denotes the application of update $u = \pm(a, b)$ to $\mathcal{A}_0$.
- Multiple updates $\{+(a_1, b_1), \ldots, +(a_n, b_n), -(a'_1, b'_1), \ldots, -(a'_m, b'_m)\}$ can be simulated by a single attack update

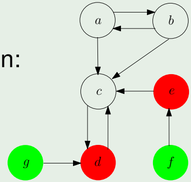
Example (Extensions/labellings after adding the attack $+(g, d)$)



grounded extension: $\{f, g\}$



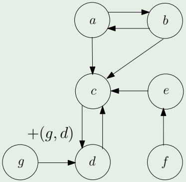
ideal extension: $\{f, g\}$



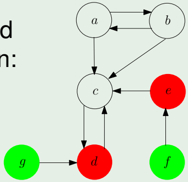
Updates

- An *update* u for an AF $\mathcal{A}_0$ consists in modifying $\mathcal{A}_0$ into an AF $\mathcal{A}$ by adding or removing arguments or attacks.
- If E_0 is an extension for $\mathcal{A}_0$ and $\mathcal{A}$ is obtained by adding (resp. removing) the set S of isolated arguments, then $E = E_0 \cup S$ (resp. $E = E_0 \setminus S$)
- We focus on the addition $+(a, b)$ and deletion $-(a, b)$ of an attack (a, b)
- $u(\mathcal{A}_0)$ denotes the application of update $u = \pm(a, b)$ to $\mathcal{A}_0$.
- Multiple updates $\{+(a_1, b_1), \ldots, +(a_n, b_n), -(a'_1, b'_1), \ldots, -(a'_m, b'_m)\}$ can be simulated by a single attack update

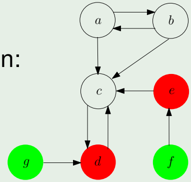
Example (Extensions/labellings after adding the attack $+(g, d)$)



grounded extension: $\{f, g\}$



ideal extension: $\{f, g\}$



Outline

1 Introduction
- Motivation
- Contributions

2 Preliminaries
- Abstract Argumentation Frameworks
- Updates

3 Algorithms
- Influenced Arguments
- Incremental Computation of Grounded Semantics
- Incremental Computation of Ideal Semantics

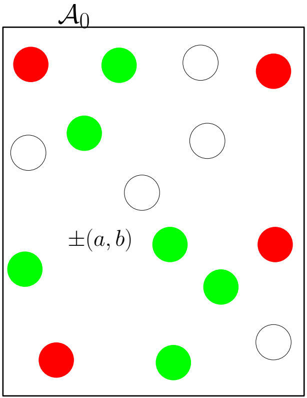
4 Experiments

5 Conclusions and future work
- References

Overview of the approach

We have an initial AF $\mathcal{A}_0$, extension E_0, and updated $u = \pm(a, b)$
Fundamental Steps:

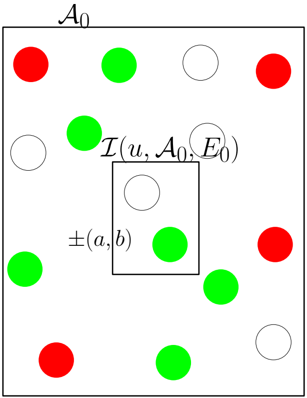
1) Compute the set of arguments of $\mathcal{A}_0$ whose status can change after performing update u (*Influenced Set* $\mathcal{I}(u, \mathcal{A}_0, E_0)$)
2) Compute the part of $\mathcal{A}_0$ induced by the influenced arguments and additional arguments containing needed information on the "external context" (*Restricted AF* $\mathcal{R}_{sem}(u, \mathcal{A}_0, E_0)$)
3) Compute the extension S_{IN} of the restricted AF using an iterative algorithm (we propose incremental algorithms for grounded and ideal semantics)
4) Combine S_{IN} with the initial extension E_0 to get the extension E of the updated AF $\mathcal{A} = u(\mathcal{A}_0)$



Overview of the approach

We have an initial AF $\mathcal{A}_0$, extension E_0, and updated $u = \pm(a, b)$
Fundamental Steps:

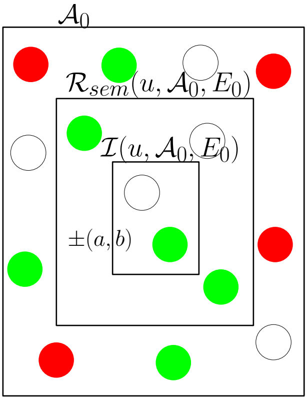
1) Compute the set of arguments of $\mathcal{A}_0$ whose status can change after performing update u (*Influenced Set* $\mathcal{I}(u, \mathcal{A}_0, E_0)$)
2) Compute the part of $\mathcal{A}_0$ induced by the influenced arguments and additional arguments containing needed information on the "external context" (*Restricted AF* $\mathcal{R}_{sem}(u, \mathcal{A}_0, E_0)$)
3) Compute the extension S_{IN} of the restricted AF using an iterative algorithm (we propose incremental algorithms for grounded and ideal semantics)
4) Combine S_{IN} with the initial extension E_0 to get the extension E of the updated AF $\mathcal{A} = u(\mathcal{A}_0)$



Overview of the approach

We have an initial AF $\mathcal{A}_0$, extension E_0, and updated $u = \pm(a, b)$
Fundamental Steps:

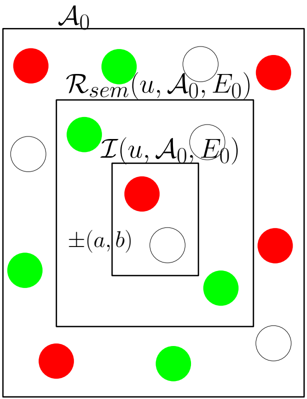
1) Compute the set of arguments of $\mathcal{A}_0$ whose status can change after performing update u (*Influenced Set* $\mathcal{I}(u, \mathcal{A}_0, E_0)$)
2) Compute the part of $\mathcal{A}_0$ induced by the influenced arguments and additional arguments containing needed information on the "external context" (*Restricted AF* $\mathcal{R}_{sem}(u, \mathcal{A}_0, E_0)$)
3) Compute the extension S_{IN} of the restricted AF using an iterative algorithm (we propose incremental algorithms for grounded and ideal semantics)
4) Combine S_{IN} with the initial extension E_0 to get the extension E of the updated AF $\mathcal{A} = u(\mathcal{A}_0)$



Overview of the approach

We have an initial AF $\mathcal{A}_0$, extension E_0, and updated $u = \pm(a, b)$
Fundamental Steps:

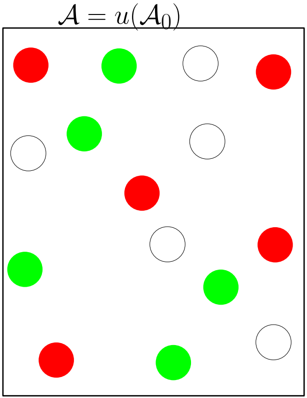
1) Compute the set of arguments of $\mathcal{A}_0$ whose status can change after performing update u (*Influenced Set* $\mathcal{I}(u, \mathcal{A}_0, E_0)$)
2) Compute the part of $\mathcal{A}_0$ induced by the influenced arguments and additional arguments containing needed information on the "external context" (*Restricted AF* $\mathcal{R}_{sem}(u, \mathcal{A}_0, E_0)$)
3) Compute the extension S_{IN} of the restricted AF using an iterative algorithm (we propose incremental algorithms for grounded and ideal semantics)
4) Combine S_{IN} with the initial extension E_0 to get the extension E of the updated AF $\mathcal{A} = u(\mathcal{A}_0)$



Overview of the approach

We have an initial AF $\mathcal{A}_0$, extension E_0, and updated $u = \pm(a, b)$
Fundamental Steps:

1) Compute the set of arguments of $\mathcal{A}_0$ whose status can change after performing update u (*Influenced Set* $\mathcal{I}(u, \mathcal{A}_0, E_0)$)
2) Compute the part of $\mathcal{A}_0$ induced by the influenced arguments and additional arguments containing needed information on the "external context" (*Restricted AF* $\mathcal{R}_{sem}(u, \mathcal{A}_0, E_0)$)
3) Compute the extension S_{IN} of the restricted AF using an iterative algorithm (we propose incremental algorithms for grounded and ideal semantics)
4) Combine S_{IN} with the initial extension E_0 to get the extension E of the updated AF $\mathcal{A} = u(\mathcal{A}_0)$



Conditions for extension/labelling preservation

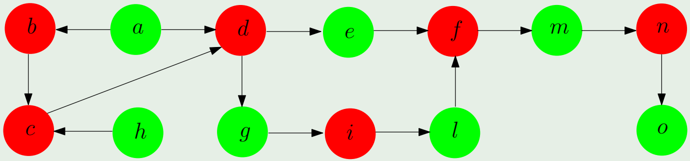
- $\mathcal{E}_{\mathcal{S}}(\mathcal{A})$ denotes the set of extensions of AF $\mathcal{A}$ according to $\mathcal{S}$

Proposition (*Addition of an attack*)

Let $u = +(a, b)$ *and* $E_0 \in \mathcal{E}_{\mathcal{S}}(\mathcal{A}_0)$ *be an extension of* $\mathcal{A}_0$ *under semantics* $\mathcal{S}$, *and* L_0 *the labelling corresponding to* E_0. *Then* $E_0 \in \mathcal{E}_{\mathcal{S}}(u(\mathcal{A}_0))$ *if*

- $\mathcal{S} \in \{co, st, gr\}$ *and one of the following conditions holds:*
  - $L_0(a) \neq$ IN *and* $L_0(b) \neq$ IN,
  - $L_0(a) =$ IN *and* $L_0(b) =$ OUT;
- $\mathcal{S} \in \{pr, ss, id\}$ *and* $L_0(b) =$ OUT.

Example (Update $+(g, f)$ does not change the initial extension E_0)



Conditions for extension/labelling preservation

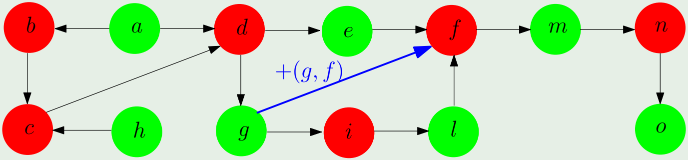
- $\mathcal{E}_\mathcal{S}(\mathcal{A})$ denotes the set of extensions of AF $\mathcal{A}$ according to $\mathcal{S}$

Proposition (*Addition of an attack*)

Let $u = +(a, b)$ and $E_0 \in \mathcal{E}_\mathcal{S}(\mathcal{A}_0)$ be an extension of $\mathcal{A}_0$ under semantics $\mathcal{S}$, and L_0 the labelling corresponding to E_0. Then $E_0 \in \mathcal{E}_\mathcal{S}(u(\mathcal{A}_0))$ if

- $\mathcal{S} \in \{\mathtt{co}, \mathtt{st}, \mathtt{gr}\}$ *and one of the following conditions holds:*
  - $L_0(a) \neq$ IN *and* $L_0(b) \neq$ IN,
  - $L_0(a) =$ IN *and* $L_0(b) =$ OUT;
- $\mathcal{S} \in \{\mathtt{pr}, \mathtt{ss}, \mathtt{id}\}$ *and* $L_0(b) =$ OUT.

Example (Update $+(g, f)$ does not change the initial extension E_0)



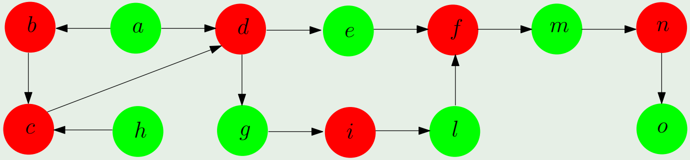
Conditions for extension/labelling preservation

- $\mathcal{E}_\mathcal{S}(\mathcal{A})$ denotes the set of extensions of AF $\mathcal{A}$ according to $\mathcal{S}$

Proposition (*Deletion of an attack*)

Let $u = -(a, b)$, $\mathcal{S} \in \{\mathtt{co}, \mathtt{pr}, \mathtt{ss}, \mathtt{st}, \mathtt{gr}\}$, and $E_0 \in \mathcal{E}_\mathcal{S}(\mathcal{A}_0)$ an extension of $\mathcal{A}_0$ under $\mathcal{S}$. Then $E_0 \in \mathcal{E}_\mathcal{S}(u(\mathcal{A}_0))$ if one of the following conditions holds:
1) $L_0(a) = \mathrm{OUT}$; 2) $L_0(a) = \mathrm{UN}$ and $L_0(b) = \mathrm{OUT}$.

Example (Update $-(c, d)$ does not change the initial extension E_0)



- In these cases we do not need to recompute the semantics

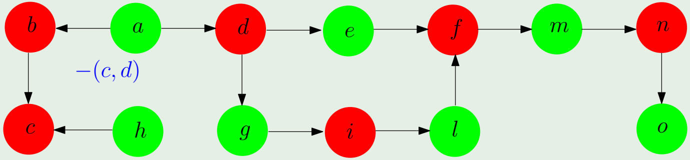
Conditions for extension/labelling preservation

- $\mathcal{E}_{\mathcal{S}}(\mathcal{A})$ denotes the set of extensions of AF $\mathcal{A}$ according to $\mathcal{S}$

Proposition (*Deletion of an attack*)

Let $u = -(a, b)$, $\mathcal{S} \in \{\texttt{co}, \texttt{pr}, \texttt{ss}, \texttt{st}, \texttt{gr}\}$, and $E_0 \in \mathcal{E}_{\mathcal{S}}(\mathcal{A}_0)$ an extension of $\mathcal{A}_0$ under $\mathcal{S}$. Then $E_0 \in \mathcal{E}_{\mathcal{S}}(u(\mathcal{A}_0))$ if one of the following conditions holds:
1) $L_0(a) = \text{OUT}$; *2) $L_0(a) = \text{UN}$ and $L_0(b) = \text{OUT}$.*

Example (Update $-(c, d)$ does not change the initial extension E_0)



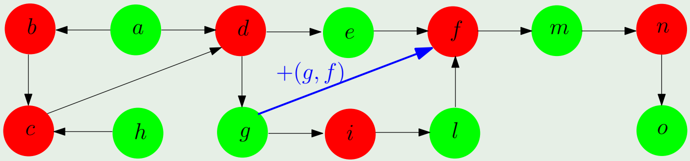
- In these cases we do not need to recompute the semantics

Influenced set: Intuition

- $\mathcal{I}(u, \mathcal{A}_0, E_0))$ is the *influenced set* of $u = \pm(a, b)$ w.r.t. $\mathcal{A}_0$ and E_0

1) if a condition for extension preservation holds, then $\mathcal{I}(u, \mathcal{A}_0, E_0) = \emptyset$
2) the status of an argument can change only if it is reachable from b (that is, $\mathcal{I}(u, \mathcal{A}_0, E_0) \subseteq Reach_{\mathcal{A}}(b)$)
3) if argument z is not reachable from b and $z \in E_0$, then also the status of the arguments attacked by z cannot change: their status remain OUT

Example (Set of arguments influenced by an update operation)



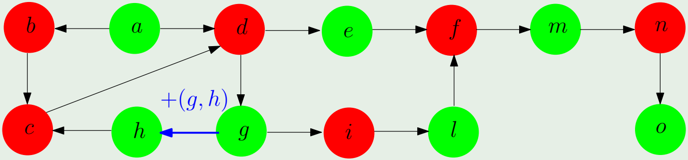
$\mathcal{I}(+(g, f), \mathcal{A}_0, E_0) = \emptyset$

Influenced set: Intuition

- $\mathcal{I}(u, \mathcal{A}_0, E_0))$ is the *influenced set* of $u = \pm(a, b)$ w.r.t. $\mathcal{A}_0$ and E_0

1) if a condition for extension preservation holds, then $\mathcal{I}(u, \mathcal{A}_0, E_0) = \emptyset$
2) the status of an argument can change only if it is reachable from b (that is, $\mathcal{I}(u, \mathcal{A}_0, E_0) \subseteq Reach_{\mathcal{A}}(b)$)
3) if argument z is not reachable from b and $z \in E_0$, then also the status of the arguments attacked by z cannot change: their status remain OUT

Example (Set of arguments influenced by an update operation)



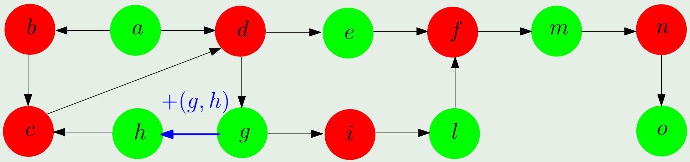
$\mathcal{I}(+(g, h), \mathcal{A}_0, E_0) \subseteq Reach_{\mathcal{A}}(h) = (A \setminus \{a, b\}) = \{c, d, e, f, g, h, i, l, m, n, o\}$
We have that $a, b \notin \mathcal{I}(+(g, h), \mathcal{A}_0, E_0)$

Influenced set: Intuition

- $\mathcal{I}(u, \mathcal{A}_0, E_0))$ is the *influenced set* of $u = \pm(a, b)$ w.r.t. $\mathcal{A}_0$ and E_0

1) if a condition for extension preservation holds, then $\mathcal{I}(u, \mathcal{A}_0, E_0) = \emptyset$
2) the status of an argument can change only if it is reachable from b (that is, $\mathcal{I}(u, \mathcal{A}_0, E_0) \subseteq Reach_{\mathcal{A}}(b)$)
3) if argument z is not reachable from b and $z \in E_0$, then also the status of the arguments attacked by z cannot change: their status remain OUT

Example (Set of arguments influenced by an update operation)



$d \notin \mathcal{I}(+(g, h), \mathcal{A}_0, E_0)$ since it is attacked by $a \in E_0$ and a is not reachable from h. Thus the arguments that can be reached only using d cannot belong to $\mathcal{I}(+(g, h), \mathcal{A}_0, E_0)$. $\rightarrow$ **The influenced set is** $\mathcal{I}(+(g, h), \mathcal{A}_0, E_0) = \{h, c\}$

Influenced set: Definition

- $\mathcal{I}(\pm(a,b),\mathcal{A}_0,E_0)$ is the set of arguments that can be reached from b without using any intermediate argument y whose status is known to be OUT because it is determined by an argument $z \in E_0$ which is not reachable from b

Definition (Influenced set)

Let $\mathcal{A} = \langle A, \Sigma \rangle$ be an AF, $u = \pm(a,b)$ an update, E an extension of $\mathcal{A}$ under a given semantics $\mathcal{S}$, and let

- $\mathcal{I}_0(u,\mathcal{A},E) = \begin{cases} \emptyset & \text{if } E \in \mathcal{E}_{\mathcal{S}}(u(\mathcal{A}))\ \ [\text{ Prop. 1/2 }] \text{ or} \\ & \exists (z,b) \in \Sigma \text{ s.t. } z \in E \wedge z \notin Reach_{\mathcal{A}}(b); \\ \{b\} & \text{otherwise}; \end{cases}$

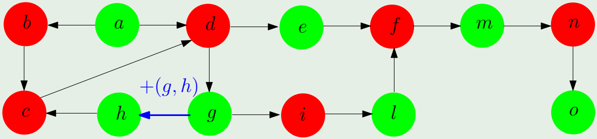
- $\mathcal{I}_{i+1}(u,\mathcal{A},E) = \mathcal{I}_i(u,\mathcal{A},E) \cup \{y \mid \exists (x,y) \in \Sigma \text{ s.t. } x \in \mathcal{I}_i(u,\mathcal{A},E) \wedge \nexists (z,y) \in \Sigma \text{ s.t. } z \in E \wedge z \notin Reach_{\mathcal{A}}(b)\}$.

The influenced set of u w.r.t. $\mathcal{A}$ and E is $\mathcal{I}(u,\mathcal{A},E) = \mathcal{I}_n(u,\mathcal{A},E)$ such that $\mathcal{I}_n(u,\mathcal{A},E) = \mathcal{I}_{n+1}(u,\mathcal{A},E)$. □

Restricted AF for grounded semantics

- Given an AF $\mathcal{A}_0$, its grounded extension E_0, and an update $u = \pm(a, b)$, the grounded semantics is recomputed for a small part of the initial AF, called restricted AF and denoted $\mathcal{R}_{\mathrm{gr}}(u, \mathcal{A}_0, E_0)$
- $\mathcal{R}_{\mathrm{gr}}(u, \mathcal{A}_0, E_0)$ consists of the subgraph of $u(\mathcal{A}_0)$ induced by $\mathcal{I}(u, \mathcal{A}_0, E_0)$
- plus additional nodes/edges representing the “external context”:
  1) if there is in $u(\mathcal{A}_0)$ an edge from a node $a \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $b \in \mathcal{I}(u, \mathcal{A}_0, E_0)$, we add edge (a, b) if the status of a is IN,
  2) if there is in $u(\mathcal{A}_0)$ an edge from a node $e \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $c \in \mathcal{I}(u, \mathcal{A}_0, E_0)$ such that e in UN, we add edge (c, c) to $\mathcal{R}_{\mathrm{gr}}(u, \mathcal{A}_0, E_0)$

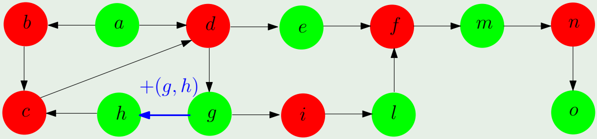
Example (Restricted AF for grounded semantics)



Restricted AF for grounded semantics

- Given an AF $\mathcal{A}_0$, its grounded extension E_0, and an update $u = \pm(a, b)$, the grounded semantics is recomputed for a small part of the initial AF, called restricted AF and denoted $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$
- $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$ consists of the subgraph of $u(\mathcal{A}_0)$ induced by $\mathcal{I}(u, \mathcal{A}_0, E_0)$
- plus additional nodes/edges representing the “external context”:
  1) if there is in $u(\mathcal{A}_0)$ an edge from a node $a \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $b \in \mathcal{I}(u, \mathcal{A}_0, E_0)$, we add edge (a, b) if the status of a is IN,
  2) if there is in $u(\mathcal{A}_0)$ an edge from a node $e \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $c \in \mathcal{I}(u, \mathcal{A}_0, E_0)$ such that e in UN, we add edge (c, c) to $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$

Example (Restricted AF for grounded semantics)

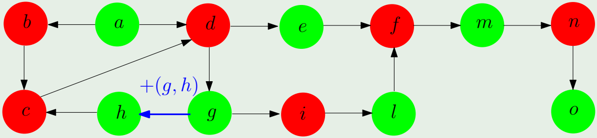


$\mathcal{I}(+(g, h), \mathcal{A}_0, E_0) = \{h, c\}$ subgraph induced by $\mathcal{I}$

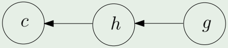
Restricted AF for grounded semantics

- Given an AF $\mathcal{A}_0$, its grounded extension E_0, and an update $u = \pm(a, b)$, the grounded semantics is recomputed for a small part of the initial AF, called restricted AF and denoted $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$
- $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$ consists of the subgraph of $u(\mathcal{A}_0)$ induced by $\mathcal{I}(u, \mathcal{A}_0, E_0)$
- plus additional nodes/edges representing the "external context":
  1) if there is in $u(\mathcal{A}_0)$ an edge from a node $a \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $b \in \mathcal{I}(u, \mathcal{A}_0, E_0)$, we add edge (a, b) if the status of a is IN,
  2) if there is in $u(\mathcal{A}_0)$ an edge from a node $e \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $c \in \mathcal{I}(u, \mathcal{A}_0, E_0)$ such that e in UN, we add edge (c, c) to $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$

Example (Restricted AF for grounded semantics)



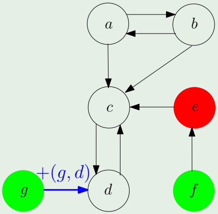
$\mathcal{I}(+(g, h), \mathcal{A}_0, E_0) = \{h, c\}$ Restricted AF:



Restricted AF for grounded semantics

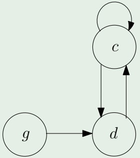
- Given an AF $\mathcal{A}_0$, its grounded extension E_0, and an update $u = \pm(a, b)$, the grounded semantics is recomputed for a small part of the initial AF, called restricted AF and denoted $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$
- $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$ consists of the subgraph of $u(\mathcal{A}_0)$ induced by $\mathcal{I}(u, \mathcal{A}_0, E_0)$
- plus additional nodes/edges representing the "external context":
  1) if there is in $u(\mathcal{A}_0)$ an edge from a node $a \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $b \in \mathcal{I}(u, \mathcal{A}_0, E_0)$, we add edge (a, b) if the status of a is IN,
  2) if there is in $u(\mathcal{A}_0)$ an edge from a node $e \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $c \in \mathcal{I}(u, \mathcal{A}_0, E_0)$ such that e in UN, we add edge (c, c) to $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$

Example (Restricted AF for grounded semantics)



$\mathcal{I}(+(g, d), \mathcal{A}_0, E_0) = \{d, c\}$

Restricted AF:



Incremental algorithm for grounded semantics

Algorithm Incr-Grounded-Sem($\mathcal{A}_0, u, E_0$)

Input: AF $\mathcal{A}_0 = \langle A_0, \Sigma_0 \rangle$, $u = \pm(a, b)$, grounded extension E_0;
Output: Revised grounded extension E

```
Let S = I(u, A_0, E_0); // Compute the influenced set
Let A_d = ⟨A_d, Σ_d⟩ = R_gr(u, A_0, E_0); // Compute the restricted AF
if (A_d = ∅) then E = E_0; // If restricted AF is empty, return the initial extension E_0
else E = (E_0 \ S) ∪ IFP(A_d, E_0 ∩ A_d); // Merge E_0 with the extension of the restricted AF
```

- Let $S = \mathcal{I}(u, \mathcal{A}_0, E_0)$; // Compute the influenced set
- Let $\mathcal{A}_d = \langle A_d, \Sigma_d \rangle = \mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$; // Compute the restricted AF
- **if** ($A_d = \emptyset$) **then** $E = E_0$; // If restricted AF is empty, return the initial extension E_0
- **else** $E = (E_0 \setminus S) \cup IFP(\mathcal{A}_d, E_0 \cap A_d)$; // Merge E_0 with the extension of the restricted AF

Function *IFP($\mathcal{A}, E_0$) // Incremental FixPoint*

Input: AF $\mathcal{A} = \langle A, \Sigma \rangle$, Extension E_0;
Output: Extension E

- $S_{\text{IN}} = \Delta_{\text{IN}} = \{ a \mid \nexists (c, a) \in \Sigma \}$; // Compute the starting set of arguments labelled IN
- **if** ($S_{\text{IN}} = \emptyset$) **return** S_{IN};
- $S_{\text{OUT}} = \Delta_{\text{OUT}} = \Delta^{+}_{\text{IN}}$; // Arguments attacked by Δ_{IN} are OUT
- **repeat**
  - $\Delta_{\text{IN}} = G(S_{\text{OUT}}, \Delta_{\text{OUT}}) \setminus S_{\text{IN}}$; // Infer new arguments that can be labelled IN
  - $\Delta_{\text{OUT}} = \Delta^{+}_{\text{IN}} \setminus S_{\text{OUT}}$; // New arguments labelled OUT
  - $S_{\text{IN}} = S_{\text{IN}} \cup \Delta_{\text{IN}}$; // Update the set of arguments labelled IN
  - $S_{\text{OUT}} = S_{\text{OUT}} \cup \Delta_{\text{OUT}}$; // ... and OUT
- **until** $\Delta_{\text{IN}} \subseteq E_0$ // Until no new labels (w.r.t. the initial labelling) are inferred
- **if** ($\Delta_{\text{IN}} = \emptyset$) **return** S_{IN};
- **else return** $S_{\text{IN}} \cup (E_0 \setminus (S_{\text{IN}} \cup S_{\text{OUT}}))$; // Merge the inferred labels with existing ones

Incremental algorithm for grounded semantics

Algorithm Incr-Grounded-Sem($\mathcal{A}_0, u, E_0$)

Input: AF $\mathcal{A}_0 = \langle A_0, \Sigma_0 \rangle$, $u = \pm(a, b)$, grounded extension E_0;
Output: Revised grounded extension E

```
Let S = I(u, A_0, E_0); // Compute the influenced set
Let A_d = ⟨A_d, Σ_d⟩ = R_gr(u, A_0, E_0); // Compute the restricted AF
if (A_d = ∅) then E = E_0; // If restricted AF is empty, return the initial extension E_0
else E = (E_0 \ S) ∪ IFP(A_d, E_0 ∩ A_d); // Merge E_0 with the extension of the restricted AF
```

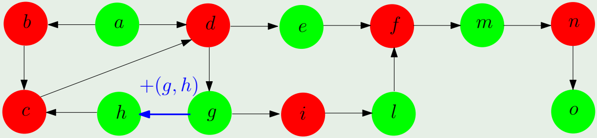
Function *IFP($\mathcal{A}, E_0$) // Incremental FixPoint*

Input: AF $\mathcal{A} = \langle A, \Sigma \rangle$, Extension E_0;
Output: Extension E

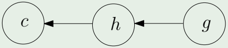
```
S_IN = Δ_IN = { a | ∄(c, a) ∈ Σ }; // Compute the starting set of arguments labelled IN
if (S_IN = ∅) return S_IN;
S_OUT = Δ_OUT = Δ_IN^+; // Arguments attacked by Δ_IN are OUT
repeat
    Δ_IN = G(S_OUT, Δ_OUT) \ S_IN; // Infer new arguments that can be labelled IN
    Δ_OUT = Δ_IN^+ \ S_OUT; // New arguments labelled OUT
    S_IN = S_IN ∪ Δ_IN; // Update the set of arguments labelled IN
    S_OUT = S_OUT ∪ Δ_OUT; // ... and OUT
until Δ_IN ⊆ E_0 // Until no new labels (w.r.t. the initial labelling) are inferred
if (Δ_IN = ∅) return S_IN;
else return S_IN ∪ (E_0 \ (S_IN ∪ S_OUT)); // Merge the inferred labels with existing ones
```

Example 1 of incremental computation

Example (From the initial extension and the update to the revised extension)



Influenced set $\mathcal{I}(+(g,h), \mathcal{A}_0, E_0) = \{h, c\}$

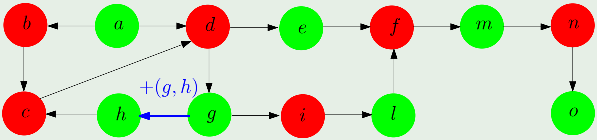
Restricted AF: 

Extension for the restricted AF:

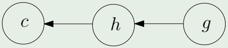
Revised extension for the updated AF:

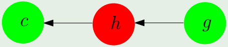
Example 1 of incremental computation

Example (From the initial extension and the update to the revised extension)



Influenced set $\mathcal{I}(+(g,h), \mathcal{A}_0, E_0) = \{h, c\}$

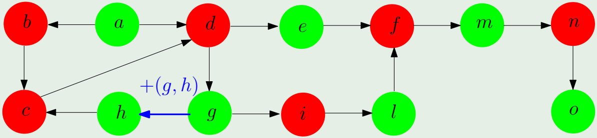
Restricted AF: 

Extension for the restricted AF: 

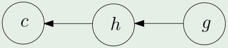
Revised extension for the updated AF:

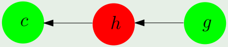
Example 1 of incremental computation

Example (From the initial extension and the update to the revised extension)

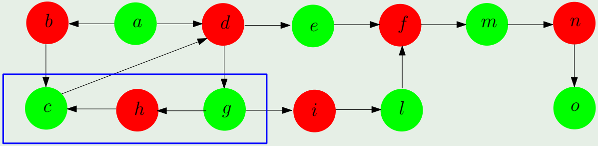


Influenced set $\mathcal{I}(+(g,h), \mathcal{A}_0, E_0) = \{h, c\}$

Restricted AF: 

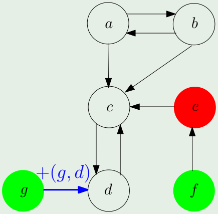
Extension for the restricted AF: 

Revised extension for the updated AF:



Example 2 of incremental computation

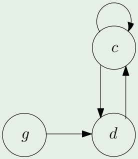
Example (From the initial extension and the update to the revised extension)



Influenced set

$\mathcal{I}(+(g,d), \mathcal{A}_0, E_0) = \{d, c\}$

Restricted AF:

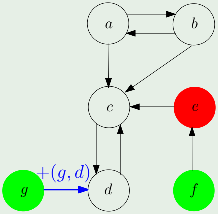


Extension for the restricted AF:

Revised extension for the updated AF:

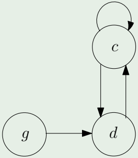
Example 2 of incremental computation

Example (From the initial extension and the update to the revised extension)

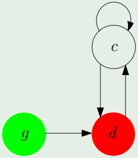


Influenced set
$\mathcal{I}(+(g,d),\mathcal{A}_0,E_0) = \{d,c\}$

Restricted AF:



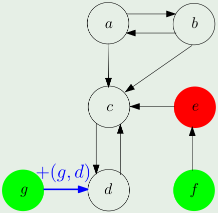
Extension for the restricted AF:



Revised extension for the updated AF:

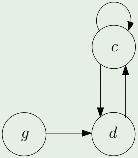
Example 2 of incremental computation

Example (From the initial extension and the update to the revised extension)

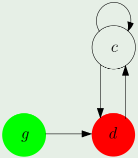


Influenced set
$\mathcal{I}(+(g,d), \mathcal{A}_0, E_0) = \{d, c\}$

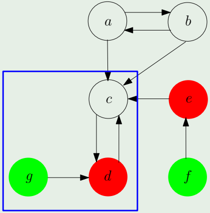
Restricted AF:



Extension for the restricted AF:



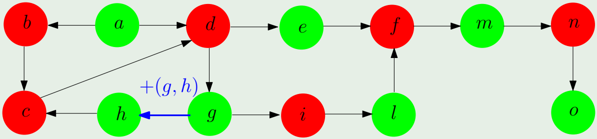
Revised extension for the updated AF:



Restricted AF for ideal semantics

- Restricted AF for ideal semantics $\mathcal{R}_{\mathtt{id}}(u, \mathcal{A}_0, E_0)$
- $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$ consists of the subgraph of $u(\mathcal{A}_0)$ induced by $\mathcal{I}(u, \mathcal{A}_0, E_0)$
- plus additional nodes/edges representing the "external context":
  1) if there is in $u(\mathcal{A}_0)$ an edge from a node $a \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $b \in \mathcal{I}(u, \mathcal{A}_0, E_0)$, we add edge (a, b) if the status of a is IN,
  2) all nodes and edges occurring in paths (of any length) ending in $\mathcal{I}(u, \mathcal{A}_0, E_0)$ whose nodes outside $\mathcal{I}(u, \mathcal{A}_0, E_0)$ are all labeled as UN.

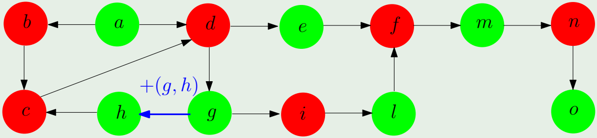
Example (Restricted AF for ideal semantics)



Restricted AF for ideal semantics

- Restricted AF for ideal semantics $\mathcal{R}_{\mathtt{id}}(u, \mathcal{A}_0, E_0)$
- $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$ consists of the subgraph of $u(\mathcal{A}_0)$ induced by $\mathcal{I}(u, \mathcal{A}_0, E_0)$
- plus additional nodes/edges representing the "external context":
  1) if there is in $u(\mathcal{A}_0)$ an edge from a node $a \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $b \in \mathcal{I}(u, \mathcal{A}_0, E_0)$, we add edge (a, b) if the status of a is IN,
  2) all nodes and edges occurring in paths (of any length) ending in $\mathcal{I}(u, \mathcal{A}_0, E_0)$ whose nodes outside $\mathcal{I}(u, \mathcal{A}_0, E_0)$ are all labeled as UN.

Example (Restricted AF for ideal semantics)

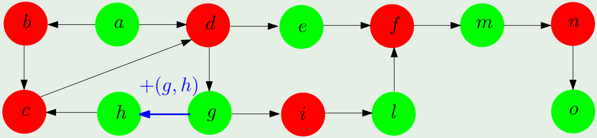


$\mathcal{I}(+(g,h), \mathcal{A}_0, E_0) = \{h, c\}$ subgraph induced by $\mathcal{I}$

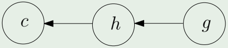
Restricted AF for ideal semantics

- Restricted AF for ideal semantics $\mathcal{R}_{\mathtt{id}}(u, \mathcal{A}_0, E_0)$
- $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$ consists of the subgraph of $u(\mathcal{A}_0)$ induced by $\mathcal{I}(u, \mathcal{A}_0, E_0)$
- plus additional nodes/edges representing the "external context":
  1) if there is in $u(\mathcal{A}_0)$ an edge from a node $a \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $b \in \mathcal{I}(u, \mathcal{A}_0, E_0)$, we add edge (a, b) if the status of a is IN,
  2) all nodes and edges occurring in paths (of any length) ending in $\mathcal{I}(u, \mathcal{A}_0, E_0)$ whose nodes outside $\mathcal{I}(u, \mathcal{A}_0, E_0)$ are all labeled as UN.

Example (Restricted AF for ideal semantics)



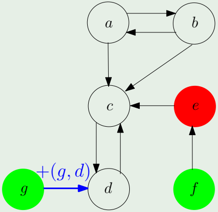
Restricted AF:



Restricted AF for ideal semantics

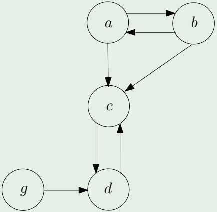
- Restricted AF for ideal semantics $\mathcal{R}_{\mathtt{id}}(u, \mathcal{A}_0, E_0)$
- $\mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E_0)$ consists of the subgraph of $u(\mathcal{A}_0)$ induced by $\mathcal{I}(u, \mathcal{A}_0, E_0)$
- plus additional nodes/edges representing the "external context":
  1) if there is in $u(\mathcal{A}_0)$ an edge from a node $a \notin \mathcal{I}(u, \mathcal{A}_0, E_0)$ to a node $b \in \mathcal{I}(u, \mathcal{A}_0, E_0)$, we add edge (a, b) if the status of a is IN,
  2) all nodes and edges occurring in paths (of any length) ending in $\mathcal{I}(u, \mathcal{A}_0, E_0)$ whose nodes outside $\mathcal{I}(u, \mathcal{A}_0, E_0)$ are all labeled as UN.

Example (Restricted AF for ideal semantics)



$\mathcal{I}(+(g, d), \mathcal{A}_0, E_0) = \{d, c\}$

Restricted AF:



Incremental algorithm for ideal semantics

Algorithm Incr-Ideal-Sem($\mathcal{A}_0, u, E_0$)

Input: AF $\mathcal{A}_0 = \langle A_0, \Sigma_0 \rangle$, $u = \pm(a, b)$, Ideal extension E_0;
Output: Revised ideal extension E;

Let $\mathcal{A} = u(\mathcal{A}_0)$;
$S = \mathcal{I}(u, \mathcal{A}_0, E_0)$; // Compute the influenced set
$E = E_0 \setminus S$; // The status of influenced arguments needs to be computed
if $(S = \emptyset)$ **then return** // If the influenced set is empty, done
while $(S \neq \emptyset)$ **do**
    $\mathcal{A}_d = \langle A_d, \Sigma_d \rangle = \mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E)$; // Compute the restricted AF for grounded semantics
    $\Delta_{\text{IN}} = IFP(\mathcal{A}_d, E \cap A_d)$; // Computed the grounded semantics
    $S = S \setminus (\Delta_{\text{IN}} \cup \Delta_{\text{IN}}^{+})$; // Remove from decided arguments
    $E = E \cup \Delta_{\text{IN}}$; // Update the extension being computed
    $\mathcal{A}_d = \mathcal{R}_{\mathtt{id}}(u, \mathcal{A}_0, E)$; // Compute the restricted AF for ideal semantics
    Select an argument $c \in S$;
    if $\exists$ successful CWS $w \in \mathcal{CW}(c, \mathcal{A}_d, E)$ **then**
        $\Delta_{\text{IN}} = PRO(w)$; // A Coherent Winning Strategy (CWS) proves whether
        $S = S \setminus (\Delta_{\text{IN}} \cup \Delta_{\text{IN}}^{+})$; // a list of arguments belong to the ideal extension
        $E = E \cup \Delta_{\text{IN}}$;
    else $S = S \setminus \{c\}$; // Otherwise, c is not in the ideal extension

Incremental algorithm for ideal semantics

Algorithm Incr-Ideal-Sem($\mathcal{A}_0, u, E_0$)

Input: AF $\mathcal{A}_0 = \langle A_0, \Sigma_0 \rangle$, $u = \pm(a, b)$, Ideal extension E_0;
Output: Revised ideal extension E;

Let $\mathcal{A} = u(\mathcal{A}_0)$;
$S = \mathcal{I}(u, \mathcal{A}_0, E_0)$; // Compute the influenced set
$E = E_0 \setminus S$; // The status of influenced arguments needs to be computed
if ($S = \emptyset$) **then return** // If the influenced set is empty, done
while ($S \neq \emptyset$) **do**
  $\mathcal{A}_d = \langle A_d, \Sigma_d \rangle = \mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E)$; // Compute the restricted AF for grounded semantics
  $\Delta_{\text{IN}} = IFP(\mathcal{A}_d, E \cap A_d)$; // Computed the grounded semantics
  $S = S \setminus (\Delta_{\text{IN}} \cup \Delta_{\text{IN}}^{+})$; // Remove from decided arguments
  $E = E \cup \Delta_{\text{IN}}$; // Update the extension being computed
  $\mathcal{A}_d = \mathcal{R}_{\mathtt{id}}(u, \mathcal{A}_0, E)$; // Compute the restricted AF for ideal semantics
  Select an argument $c \in S$;
  if $\exists$ successful CWS $w \in \mathcal{CW}(c, \mathcal{A}_d, E)$ **then**
    $\Delta_{\text{IN}} = PRO(w)$; // A Coherent Winning Strategy (CWS) proves whether
    $S = S \setminus (\Delta_{\text{IN}} \cup \Delta_{\text{IN}}^{+})$; // a list of arguments belong to the ideal extension
    $E = E \cup \Delta_{\text{IN}}$;
  else $S = S \setminus \{c\}$; // Otherwise, c is not in the ideal extension

Incremental algorithm for ideal semantics

Algorithm Incr-Ideal-Sem($\mathcal{A}_0, u, E_0$)

Input: AF $\mathcal{A}_0 = \langle A_0, \Sigma_0 \rangle$, $u = \pm(a, b)$, Ideal extension E_0;
Output: Revised ideal extension E;

Let $\mathcal{A} = u(\mathcal{A}_0)$;
$S = \mathcal{I}(u, \mathcal{A}_0, E_0)$; // Compute the influenced set
$E = E_0 \setminus S$; // The status of influenced arguments needs to be computed
if $(S = \emptyset)$ **then return** // If the influenced set is empty, done
while $(S \neq \emptyset)$ **do**
    $\mathcal{A}_d = \langle A_d, \Sigma_d \rangle = \mathcal{R}_{\mathtt{gr}}(u, \mathcal{A}_0, E)$; // Compute the restricted AF for grounded semantics
    $\Delta_{\text{IN}} = IFP(\mathcal{A}_d, E \cap A_d)$; // Computed the grounded semantics
    $S = S \setminus (\Delta_{\text{IN}} \cup \Delta_{\text{IN}}^{+})$; // Remove from decided arguments
    $E = E \cup \Delta_{\text{IN}}$; // Update the extension being computed
    $\mathcal{A}_d = \mathcal{R}_{\mathtt{id}}(u, \mathcal{A}_0, E)$; // Compute the restricted AF for ideal semantics
    Select an argument $c \in S$;
    if $\exists$ successful CWS $w \in \mathcal{CW}(c, \mathcal{A}_d, E)$ **then**
        $\Delta_{\text{IN}} = PRO(w)$; // A Coherent Winning Strategy (CWS) proves whether
        $S = S \setminus (\Delta_{\text{IN}} \cup \Delta_{\text{IN}}^{+})$; // a list of arguments belong to the ideal extension
        $E = E \cup \Delta_{\text{IN}}$;
    else $S = S \setminus \{c\}$; // Otherwise, c is not in the ideal extension

Outline

1 Introduction
- Motivation
- Contributions

2 Preliminaries
- Abstract Argumentation Frameworks
- Updates

3 Algorithms
- Influenced Arguments
- Incremental Computation of Grounded Semantics
- Incremental Computation of Ideal Semantics

4 Experiments

5 Conclusions and future work
- References

Datasets and algorithms

Datasets

- For grounded semantics, datasets from ICCMA (International Competition on Computational Models of Argumentation)
  - `REAL` : 19 AFs $\langle A_0, \Sigma_0 \rangle$ with $|A_0| \in [5K, 100K]$ and $|\Sigma_0| \in [7K, 143K]$
  - `SYN1` : 24 AFs $\langle A_0, \Sigma_0 \rangle$ with $|A_0| \in [1K, 4K]$ and $|\Sigma_0| \in [14K, 172K]$
- For ideal semantics, `SYN2` consists of 20 AFs with $|A_0| \in \{50, 75, \ldots 175\}$

Algorithms:

- ***BaseG*** computes the grounded extension E of the updated AF $u(\mathcal{A}_0)$ from scratch: it finds the fixpoint of the characteristic function of an AF as implemented in the libraries of the *Tweety* Project
- ***BaseI*** computes the ideal extension E of the updated AF $u(\mathcal{A}_0)$ from scratch: it uses the algorithm implemented by Dung-O-Matic engine
- *Incr-Grounded-Sem* (***IncrG***) incrementally computes the grounded extension E starting from E_0 and the update
- *Incr-Ideal-Sem* (***IncrI***) incrementally computes the ideal extension E starting from E_0 and the update

Datasets and algorithms

Datasets

- For grounded semantics, datasets from ICCMA (International Competition on Computational Models of Argumentation)
  - `REAL` : 19 AFs $\langle A_0, \Sigma_0 \rangle$ with $|A_0| \in [5K, 100K]$ and $|\Sigma_0| \in [7K, 143K]$
  - `SYN1` : 24 AFs $\langle A_0, \Sigma_0 \rangle$ with $|A_0| \in [1K, 4K]$ and $|\Sigma_0| \in [14K, 172K]$
- For ideal semantics, `SYN2` consists of 20 AFs with $|A_0| \in \{50, 75, \ldots 175\}$

Algorithms:

- ***BaseG*** computes the grounded extension E of the updated AF $u(\mathcal{A}_0)$ from scratch: it finds the fixpoint of the characteristic function of an AF as implemented in the libraries of the *Tweety* Project
- ***BaseI*** computes the ideal extension E of the updated AF $u(\mathcal{A}_0)$ from scratch: it uses the algorithm implemented by Dung-O-Matic engine
- *Incr-Grounded-Sem* (***IncrG***) incrementally computes the grounded extension E starting from E_0 and the update
- *Incr-Ideal-Sem* (***IncrI***) incrementally computes the ideal extension E starting from E_0 and the update

Datasets and algorithms

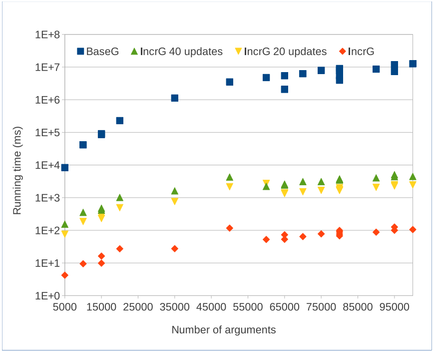
Datasets

- For grounded semantics, datasets from ICCMA (International Competition on Computational Models of Argumentation)
  - `REAL` : 19 AFs $\langle A_0, \Sigma_0 \rangle$ with $|A_0| \in [5K, 100K]$ and $|\Sigma_0| \in [7K, 143K]$
  - `SYN1` : 24 AFs $\langle A_0, \Sigma_0 \rangle$ with $|A_0| \in [1K, 4K]$ and $|\Sigma_0| \in [14K, 172K]$
- For ideal semantics, `SYN2` consists of 20 AFs with $|A_0| \in \{50, 75, \ldots 175\}$

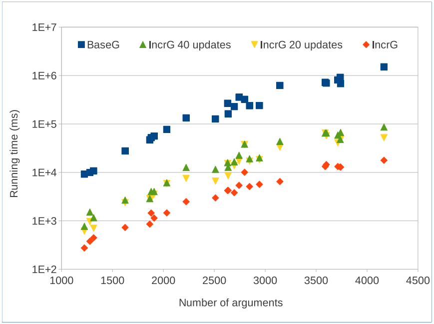
Algorithms:

- ***BaseG*** computes the grounded extension E of the updated AF $u(\mathcal{A}_0)$ from scratch: it finds the fixpoint of the characteristic function of an AF as implemented in the libraries of the *Tweety* Project
- ***BaseI*** computes the ideal extension E of the updated AF $u(\mathcal{A}_0)$ from scratch: it uses the algorithm implemented by Dung-O-Matic engine
- *Incr-Grounded-Sem* (***IncrG***) incrementally computes the grounded extension E starting from E_0 and the update
- *Incr-Ideal-Sem* (***IncrI***) incrementally computes the ideal extension E starting from E_0 and the update

Experiments for grounded semantics



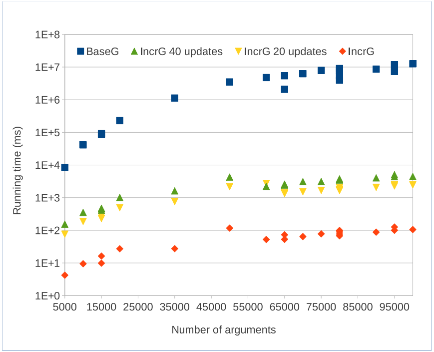
Run times (ms) of *BaseG* and *IncrG* for 1, 20, and 40 updates over REAL



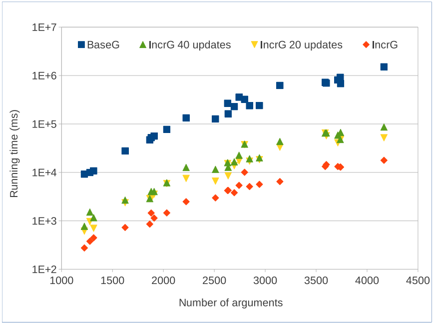
Run times (ms) of *BaseG* and *IncrG* for 1, 20, and 40 updates over SYN1

- *IncrG* and *IncrI* compute extensions of AFs updated by a set U of (simultaneous) updates by reducing the application of $U = \{+(a_1, b_1), \ldots, +(a_n, b_n), -(a'_1, b'_1), \ldots, -(a'_m, b'_m)\}$ on AF $\mathcal{A}_0$ to the application of a single attack update on an AF obtained from $\mathcal{A}_0$

Experiments for grounded semantics



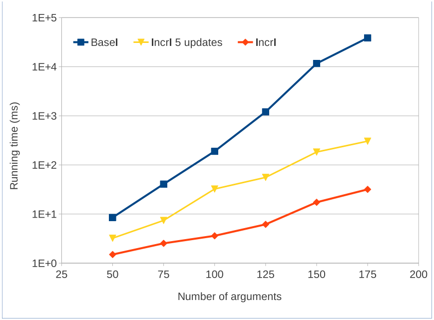
Run times (ms) of *BaseG* and *IncrG* for 1, 20, and 40 updates over `REAL`



Run times (ms) of *BaseG* and *IncrG* for 1, 20, and 40 updates over `SYN1`

- *IncrG* and *IncrI* compute extensions of AFs updated by a set U of (simultaneous) updates by reducing the application of $U = \{+(a_1, b_1), \ldots, +(a_n, b_n), -(a'_1, b'_1), \ldots, -(a'_m, b'_m)\}$ on AF $\mathcal{A}_0$ to the application of a single attack update on an AF obtained from $\mathcal{A}_0$

Experiments for ideal semantics



Run times (ms) of *BaseI* and *IncrI*
for 1 and 5 updates over SYN2

- Linear improvements for grounded semantics
- Exponential improvements for ideal semantics (whose computation from scratch is exponential)

Outline

1. Introduction
   - Motivation
   - Contributions
2. Preliminaries
   - Abstract Argumentation Frameworks
   - Updates
3. Algorithms
   - Influenced Arguments
   - Incremental Computation of Grounded Semantics
   - Incremental Computation of Ideal Semantics
4. Experiments
5. Conclusions and future work
   - References

Conclusions and future work

- We presented two incremental algorithms for computing deterministic extensions of updated AFs
- The algorithms exploit the initial extension of an AF for computing the set of arguments influenced by an update,
- and for detecting early termination conditions during the recomputation of the status of the arguments.
- The technique can be used in the case of general multiple updates.
- The experiments showed that the incremental computation outperforms that of the base (non-incremental) computation
- The definition of influenced set substantially restricts the portion of the AF to be analysed for recomputing the semantics after an update.
- Future work: application of the technique to other (multiple status) semantics.

Thank you!

... any question?

Selected References

Phan Minh Dung.
On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games.
Artif. Intell., 77(2):321–358, 1995.

Bei Shui Liao, Li Jin, Robert C. Koons.
Dynamics of argumentation systems: A division-based method.
Artif. Intell., 175(11), 1790–1814, (2011).

Baroni, P., Giacomin, M., Liao, B.
On topology-related properties of abstract argumentation semantics. A correction and extension to dynamics of argumentation systems: A division-based method.
Artificial Intelligence 212, 104–115 (2014)