# Computing Stable Extensions of Argumentation Frameworks using Formal Concept Analysis

Sergei Obiedkov[1] Barış Sertkaya[2]

[1]KBS Group, TU Dresden

[2]Frankfurt University of Applied Sciences

# Outline

Formal Concept Analysis (FCA)

Argumentation frameworks in FCA terms

Enumerating stable extensions with FCA algorithms

Experimental evaluation

# Formal Concept Analysis
(Ganter and Wille 1999)

## Formal context $\mathbb{K} = (G, M, I)$

- a set of objects $G$
- a set of attributes $M$
- objects are described with attributes: the binary relation $I \subseteq G \times M$

# Formal Concept Analysis

(Ganter and Wille 1999)

## Formal context $\mathbb{K} = (G, M, I)$

- a set of objects $G$
- a set of attributes $M$
- objects are described with attributes: the binary relation $I \subseteq G \times M$

## Derivation operators

For $A \subseteq G$ and $B \subseteq M$:

- $A^{\uparrow} = \{m \in M \mid \forall g \in A\colon (g, m) \in I\}$
- $B^{\downarrow} = \{g \in G \mid \forall m \in B\colon (g, m) \in I\}$

# Formal Concept Analysis

(Ganter and Wille 1999)

## Derivation operators

For $A \subseteq G$ and $B \subseteq M$:

- $A^{\uparrow} = \{m \in M \mid \forall g \in A \colon (g, m) \in I\}$
- $B^{\downarrow} = \{g \in G \mid \forall m \in B \colon (g, m) \in I\}$

## Formal concept $(A, B)$

- $A \subseteq G$
- $A^{\uparrow} = B$
- $B \subseteq M$
- $B^{\downarrow} = A$

$A$ is concept extent and $B$ is concept intent.

$(A, B) \leq (C, D) \Leftrightarrow A \subseteq C \ (\Leftrightarrow D \subseteq B)$

The concept set of the context $\mathbb{K}$ forms a lattice $\underline{\mathfrak{B}}(\mathbb{K})$.

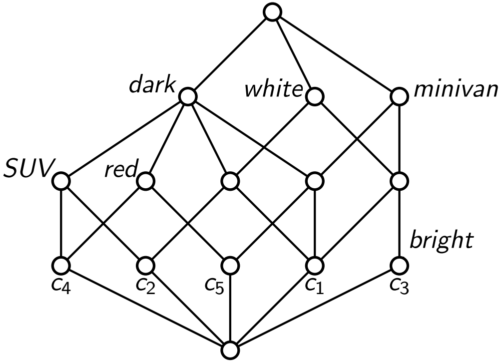
# Example

## adapted from (Brafman and Domshlak 2009)

Context

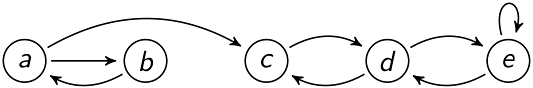
| | minivan | SUV | red exterior | white exterior | bright interior | dark interior |
|---|---|---|---|---|---|---|
| $c_1$ | × | | | × | | × |
| $c_2$ | | × | | × | | × |
| $c_3$ | × | | | × | × | |
| $c_4$ | | × | × | | | × |
| $c_5$ | × | | × | | | × |

Concept lattice



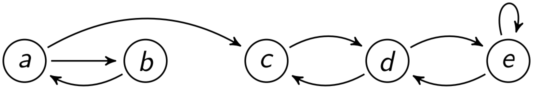
# Argumentation framework (Dung 1995)

An argumentation framework is a directed graph $F = (A, R)$, where $A$ is a finite set of arguments and $R \subseteq A \times A$ is the attack relation.



# Argumentation framework (Dung 1995)

An argumentation framework is a directed graph $F = (A, R)$, where $A$ is a finite set of arguments and $R \subseteq A \times A$ is the attack relation.

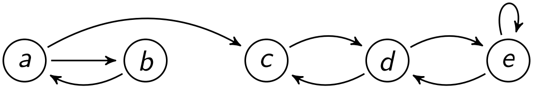



The induced formal context of $(A, R)$ is $\mathbb{K}(A, R) = (A, A, (A \times A) \setminus R)$.

| | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| $a$ | × | | | × | × |
| $b$ | | × | × | × | × |
| $c$ | × | × | × | | × |
| $d$ | × | × | | × | |
| $e$ | × | × | × | | |

# Argumentation framework (Dung 1995)

An argumentation framework is a directed graph $F = (A, R)$, where $A$ is a finite set of arguments and $R \subseteq A \times A$ is the attack relation.




$S^{\uparrow} = \{t \in A \mid \forall s \in S\colon (s,t) \notin R\}$
is the set of arguments not attacked by $S$

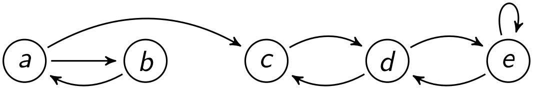
$$\{a, b\}^{\uparrow} = \{d, e\}$$

The induced formal context of $(A, R)$ is $\mathbb{K}(A,R) = (A, A, (A \times A) \setminus R)$.

| | a | b | c | d | e |
|---|---|---|---|---|---|
| a | × | | | × | × |
| b | | × | × | × | × |
| c | × | × | × | | × |
| d | × | × | | × | |
| e | × | × | × | | |

# Argumentation framework (Dung 1995)

An argumentation framework is a directed graph $F = (A, R)$, where $A$ is a finite set of arguments and $R \subseteq A \times A$ is the attack relation.




The induced formal context of $(A, R)$ is $\mathbb{K}(A, R) = (A, A, (A \times A) \setminus R)$.

| | a | b | c | d | e |
|---|---|---|---|---|---|
| a | × | | | × | × |
| b | | × | × | × | × |
| c | × | × | × | | × |
| d | × | × | | × | |
| e | × | × | × | | |

$S^{\uparrow} = \{t \in A \mid \forall s \in S\colon (s, t) \notin R\}$
is the set of arguments not attacked by $S$

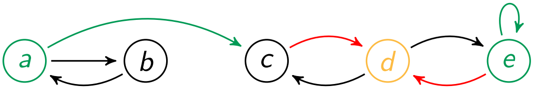
$$\{a, b\}^{\uparrow} = \{d, e\}$$

$S^{\downarrow} = \{t \in A \mid \forall s \in S\colon (t, s) \notin R\}$
is the set of arguments that do not attack $S$

$$\{a, b\}^{\downarrow} = \{c, d, e\}$$

# Defending

$S^{\uparrow}$ are the arguments not attacked by $S$




$\{a, e\}$ defends $d$

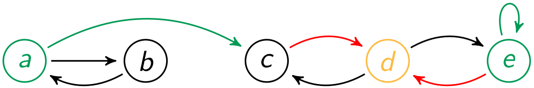
$S \subseteq A$ defends $x \in A$ if every argument attacking $x$ is attacked by $S$

$S^{\downarrow}$ are the arguments not attacking $S$

| | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| $a$ | × | | | × | × |
| $b$ | | × | × | × | × |
| $c$ | × | × | × | | × |
| $d$ | × | × | | × | |
| $e$ | × | × | × | | |

# Defending

$S^{\uparrow}$ are the arguments not attacked by $S$




$\{a, e\}$ defends $d$

$S \subseteq A$ defends $x \in A$ if every argument attacking $x$ is attacked by $S$

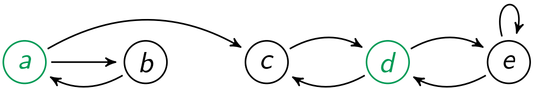
$S^{\downarrow}$ are the arguments not attacking $S$

| | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| $a$ | × | | | × | × |
| $b$ | | × | × | × | × |
| $c$ | × | × | × | | × |
| $d$ | × | × | | × | |
| $e$ | × | × | × | | |

$\iff$ $S^{\uparrow} \subseteq \{x\}^{\downarrow}$

# Conflict-free sets

$S^{\uparrow}$ are the arguments not attacked by $S$




$\{a, d\}$ is conflict-free

$S \subseteq A$ is conflict-free if $S$ does not attack any of its elements

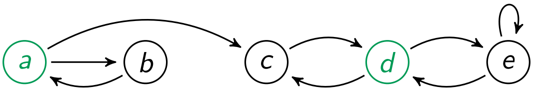
$S^{\downarrow}$ are the arguments not attacking $S$

| | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| $a$ | $\times$ | | | $\times$ | $\times$ |
| $b$ | | $\times$ | $\times$ | $\times$ | $\times$ |
| $c$ | $\times$ | $\times$ | $\times$ | | $\times$ |
| $d$ | $\times$ | $\times$ | | $\times$ | |
| $e$ | $\times$ | $\times$ | $\times$ | | |

# Conflict-free sets

$S^{\uparrow}$ are the arguments not attacked by $S$

$S^{\downarrow}$ are the arguments not attacking $S$




$\{a, d\}$ is conflict-free

|   | a | b | c | d | e |
|---|---|---|---|---|---|
| a | × |   |   | × | × |
| b |   | × | × | × | × |
| c | × | × | × |   | × |
| d | × | × |   | × |   |
| e | × | × | × |   |   |

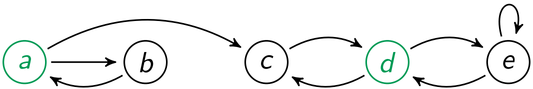
$S \subseteq A$ is conflict-free if $S$ does not attack any of its elements

$\Longleftrightarrow \quad S \subseteq S^{\uparrow} \quad \Longleftrightarrow \quad S \subseteq S^{\downarrow}$

# Conflict-free sets

$S^{\uparrow}$ are the arguments not attacked by $S$

$S^{\downarrow}$ are the arguments not attacking $S$




$\{a, d\}$ is conflict-free

| | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| $a$ | × | | | × | × |
| $b$ | | × | × | × | × |
| $c$ | × | × | × | | × |
| $d$ | × | × | | × | |
| $e$ | × | × | × | | |

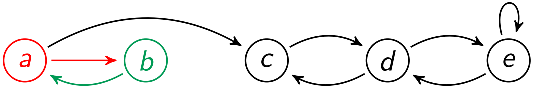
$S \subseteq A$ is conflict-free if $S$ does not attack any of its elements $\iff S \subseteq S^{\uparrow} \iff S \subseteq S^{\downarrow}$

$S \subseteq A$ is a maximal conflict-free set $\iff S = S^{\uparrow} \cap S^{\downarrow} \cap \{x \in A \mid x \in \{x\}^{\downarrow}\}$

# Admissible and preferred extensions

$S^{\uparrow}$ are the arguments not attacked by $S$

$S^{\downarrow}$ are the arguments not attacking $S$



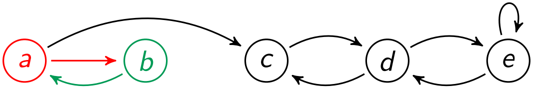

$\{b\}$ is admissible

| | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| $a$ | × | | | × | × |
| $b$ | | × | × | × | × |
| $c$ | × | × | × | | × |
| $d$ | × | × | | × | |
| $e$ | × | × | × | | |

$S \subseteq A$ is admissible if $S$ is conflict-free and $S$ defends all its elements

# Admissible and preferred extensions

$S^{\uparrow}$ are the arguments not attacked by $S$ $S^{\downarrow}$ are the arguments not attacking $S$




$\{b\}$ is admissible

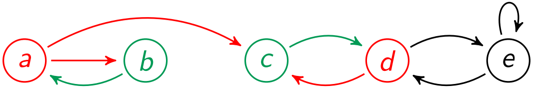
| | a | b | c | d | e |
|---|---|---|---|---|---|
| a | × | | | × | × |
| b | | × | × | × | × |
| c | × | × | × | | × |
| d | × | × | | × | |
| e | × | × | × | | |

$S \subseteq A$ is admissible if $S$ is conflict-free and $S$ defends all its elements $\iff$ $S \subseteq S^{\uparrow} \subseteq S^{\downarrow}$

# Admissible and preferred extensions

$S^{\uparrow}$ are the arguments not attacked by $S$

$S^{\downarrow}$ are the arguments not attacking $S$




$\{b, c\}$ is preferred

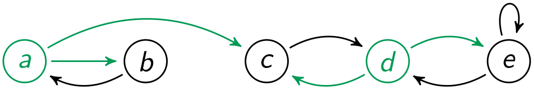
| | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| $a$ | × | | | × | × |
| $b$ | | × | × | × | × |
| $c$ | × | × | × | | × |
| $d$ | × | × | | × | |
| $e$ | × | × | × | | |

$S \subseteq A$ is admissible if $S$ is conflict-free and $S$ defends all its elements $\iff$ $S \subseteq S^{\uparrow} \subseteq S^{\downarrow}$

$S \subseteq A$ is preferred if it is a maximal admissible extension $\implies$ $S = S^{\downarrow\uparrow}$ is a concept intent

# Stable extensions

$S^{\uparrow}$ are the arguments not attacked by $S$




$\{a, d\}$ is stable

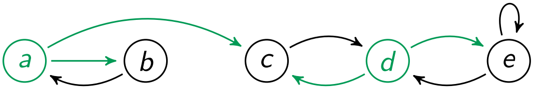
$S \subseteq A$ is stable if $S$ is conflict-free and $S$ attacks every $a \in A \setminus S$

$S^{\downarrow}$ are the arguments not attacking $S$

| | *a* | *b* | *c* | *d* | *e* |
|---|---|---|---|---|---|
| *a* | × | | | × | × |
| *b* | | × | × | × | × |
| *c* | × | × | × | | × |
| *d* | × | × | | × | |
| *e* | × | × | × | | |

# Stable extensions

$S^{\uparrow}$ are the arguments not attacked by $S$ $\qquad$ $S^{\downarrow}$ are the arguments not attacking $S$



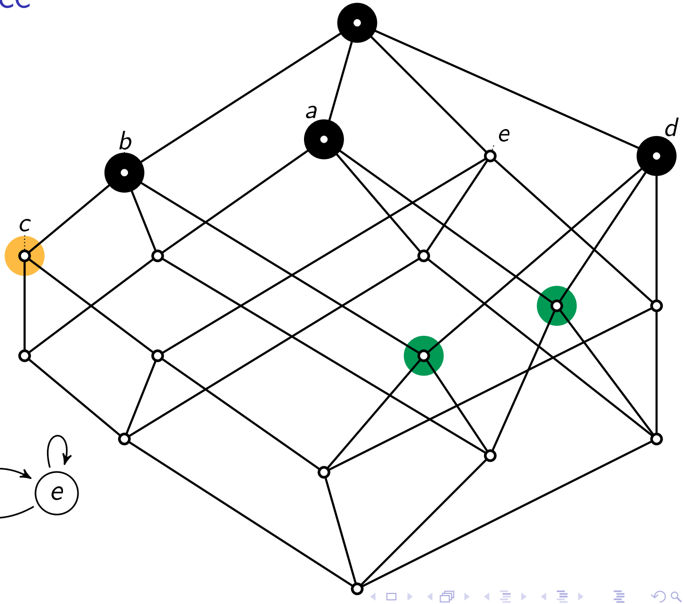

$\{a, d\}$ is stable

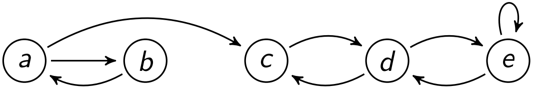
| | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| $a$ | × | | | × | × |
| $b$ | | × | × | × | × |
| $c$ | × | × | × | | × |
| $d$ | × | × | | × | |
| $e$ | × | × | × | | |

$S \subseteq A$ is stable if $S$ is conflict-free and $S$ attacks every $a \in A \setminus S$ $\iff$ $S^{\uparrow} = S$

# Extensions in the concept lattice

- stable extensions
- other preferred extensions
- other conflict-free intents



# Lattice-construction algorithms for enumeration of stable extensions

- Conflict-free intents form an order filter in the concept lattice.
- Stable extensions form an antichain.

- We use FCA algorithms to enumerate all conflict-free concept intents.
- We prune a computation branch when we encounter a maximal conflict-free intent.
- If such intent is stable, we output it.

# Lattice-construction algorithms for enumeration of stable extensions

We adapted two algorithms:

### Next Closure (Ganter 1984)

- Enumerates all concept intents with a polynomial delay.
- Lists intents contained in intent $S$ before it produces $S$.
- Needs memory linear in the number of arguments.

### Incremental algorithm (Norris 1978)

- Processes attributes one by one.
- Stores all generated concepts to ease generation of new concepts.
- May need exponential amount of memory, but, when it is available, is usually fast.

# Experiments

- We ran experiments for the task of counting stable extensions (CE-ST).
  - Results for the task of finding a single extension are in the paper.
- We used random Erdős–Rényi–Gilbert graphs ($G(n, p)$ model) for testing.
- For comparison, we used the following tools from ICCMA 2021:
  - A-Folio-DPDB
  - PYGLAF
  - $\mu$-toksia
- Experimental setup: Ubuntu Linux / 32 core-CPU / 2.9 GHz / 256 GB
- Time limit: 600 sec

# Experiments

## $G(1000, p)$, average time in seconds

| $p =$ | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 0.98 |
|---|---|---|---|---|---|---|
| PYGLAF | – | – | – | – | – | – |
| $\mu$-toksia | – | – | – | 155 | 131 | 183 |
| A-Folio-DPDB | – | – | 114 | 30.2 | 17.7 | 12.1 |
| Next Closure | 435 | 31.4 | 3.89 | 0.72 | 0.24 | 0.16 |
| Norris | 263 | 19.4 | 1.93 | 0.31 | 0.15 | 0.14 |

None of the tools terminated within limit for $p \leq 0.4$.

# Experiments

$G(n, p)$, average time in seconds

| | $n = 5000$ | | | $n = 10000$ | | | $n = 20000$ | | |
|---|---|---|---|---|---|---|---|---|---|
| $p =$ | 0.8 | 0.9 | 0.98 | 0.8 | 0.9 | 0.98 | 0.8 | 0.9 | 0.98 |
| PYGLAF | – | – | 287 | – | – | – | – | – | – |
| $\mu$-toksia | – | – | – | – | – | – | – | – | – |
| A-Folio-DPDB | – | – | – | – | – | – | – | – | – |
| Next Closure | 405 | 35.3 | 6.03 | – | 510 | 39.4 | – | – | 589 |
| Norris | 303 | 25.2 | 4.42 | – | 360 | 25.6 | – | – | 167 |

# Experiments

$G(n, p)$, average time in seconds

| | $n = 50$ | | | $n = 250$ | | | $n = 500$ | | |
|---|---|---|---|---|---|---|---|---|---|
| $p =$ | 0.01 | 0.2 | 0.5 | 0.01 | 0.2 | 0.5 | 0.01 | 0.2 | 0.5 |
| PYGLAF | 0.07 | 0.04 | 0.04 | 0.08 | 40.6 | 3.86 | 0.08 | – | 73 |
| $\mu$-toksia | 0.01 | 0.005 | 0.01 | 0.01 | 60.3 | 8.82 | 0.32 | – | 116 |
| A-Folio-DPDB | 4.84 | 4.81 | 4.82 | 5 | 40.4 | 6.73 | 5.08 | – | 52.5 |
| Next Closure | 0.004 | 0.03 | 0.003 | – | – | 0.35 | – | – | 10.1 |
| Norris | 0.003 | 0.02 | 0.003 | – | – | 0.13 | – | – | 5.29 |

# Conclusion

## The main takeaway

FCA algorithms are efficient for the enumeration of stable extensions in dense frameworks (which induce sparse contexts with relatively few concepts and even fewer conflict-free intents).

# Conclusion

## The main takeaway

FCA algorithms are efficient for the enumeration of stable extensions in dense frameworks (which induce sparse contexts with relatively few concepts and even fewer conflict-free intents).

## Further work

- Adapt the algorithms to other semantics
- Test the algorithms on differently generated frameworks
- Improve the algorithms, e.g., using ordering heuristics
- Adapt other FCA algorithms to argumentation tasks
- Compare with other lattice-based approaches
  - Elaroussi, M., Nourine, L. & Radjef, M.S. Lattice point of view for argumentation framework. *Ann Math Artif Intell* (2023)