dynPARTIX - A Dynamic Programming Reasoner for Abstract Argumentation[◇]

INAP 2011, Vienna

Wolfgang Dvořák, Michael Morak, Clemens Nopp, and Stefan Woltran

Institute of Information Systems,
Vienna University of Technology

September 28, 2011

[◇] Supported by the Vienna Science and Technology Fund (WWTF) under grant ICT08-028, by the Austrian Science Fund (FWF) under grant P20704-N18, and by the Vienna University of Technology program "Innovative Ideas".

Motivation

- Increasing interest for reasoning in argumentation frameworks (AFs).
- Most reasoning tasks are highly computationally intractable.
- As AFs can be considered as graphs there is broad range of graph parameters we can consider to identify tractable fragments.
- The popular graph parameter tree-width allows for so called fixed-parameter tractable algorithms [Dunne, 2007; Dvořák et al., 2010].

Fixed-Parameter Tractability

- Often computational costs primarily depend on some problem parameters rather than on the mere size of the instances.
- Many hard problems become tractable if some problem parameter is fixed or bounded by a fixed constant.
- In the arena of graphs an important parameter is tree-width, which measures the "tree-likeness" of a graph. The tree-width has served as the key to many fixed-parameter tractability (FPT) results.

Argumentation Frameworks

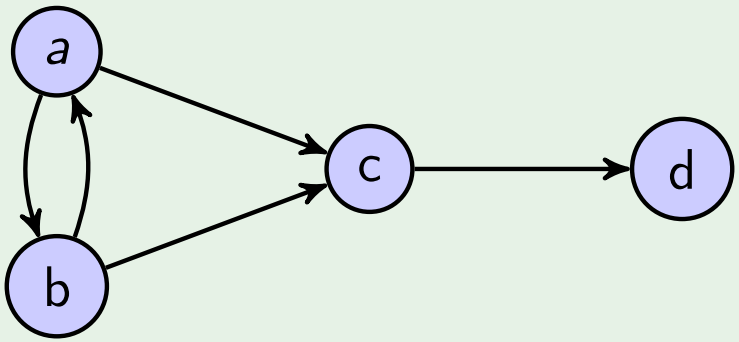
Definition

An argumentation framework (AF) is a pair (A, R) where

- A is a set of arguments
- $R \subseteq A \times A$ is a relation representing "attacks"

Example

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

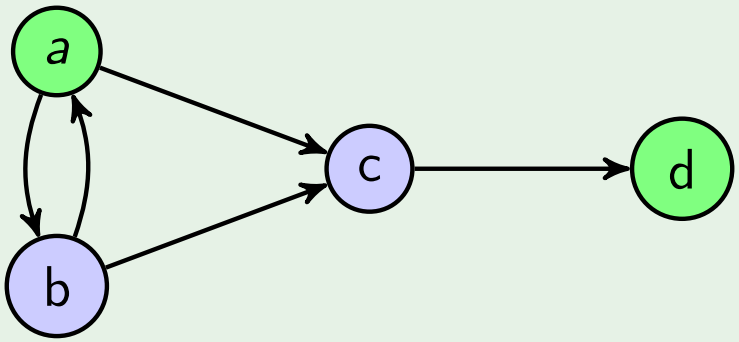


Argumentation Semantics

Conflict-Free Sets

Given an AF $F = (A, R)$.
A set $S \subseteq A$ is conflict-free in F, if, for each $a, b \in S$, $(a, b) \notin R$.

Example



$\mathcal{E}_{cf}(F) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{a, d\}, \{b, d\}\}$

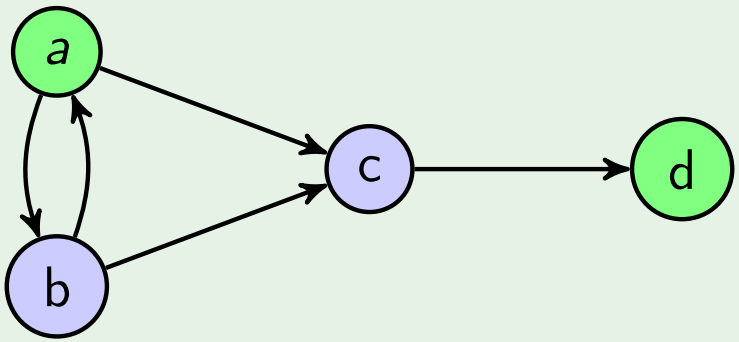
Argumentation Semantics (ctd.)

Admissible Sets

Given an AF $F = (A, R)$. A set $S \subseteq A$ is admissible in F, if

- S is conflict-free in F
- each $a \in S$ is defended by S in F
  - $a \in A$ is defended by S in F, if for each $b \in A$ with $(b, a) \in R$, there exists a $c \in S$, such that $(c, b) \in R$.

Example



$\mathcal{E}_{adm}(F) = \{\emptyset, \{a\}, \{b\}, \{\cancel{c}\}, \{a, d\}, \{b, d\}\}$

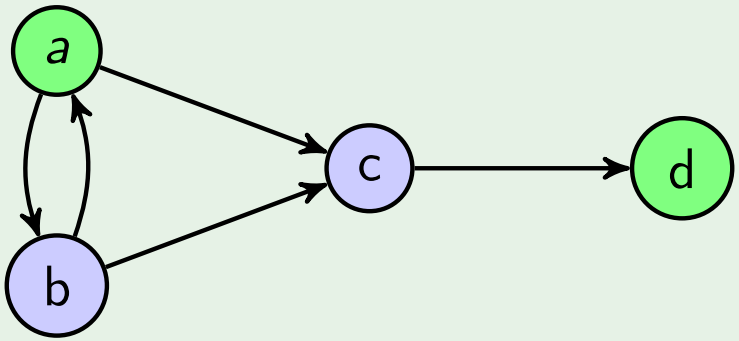
Argumentation Semantics (ctd.)

Preferred Extensions

Given an AF $F = (A, R)$. A set $S \subseteq A$ is a preferred extension of F, if

- S is admissible in F
- for each $T \subseteq A$ admissible in F, $S \not\subset T$

Example



$\mathcal{E}_{pr}(F) = \{\sout{\emptyset}, \sout{\{a\}}, \sout{\{b\}}, \{a, d\}, \{b, d\}\}$

Reasoning Problems

Credulous Acceptance

Given an AF $F = (A, R)$ and an argument $x \in A$.
Is x in at least one preferred extension ?

For credulous acceptance it suffices to consider admissible extensions.

Skeptical Acceptance

Given an AF $F = (A, R)$ and an argument $x \in A$.
Is x in every preferred extension ?

Reasoning Problems

Credulous Acceptance

Given an AF $F = (A, R)$ and an argument $x \in A$.
Is x in at least one preferred extension ?

For credulous acceptance it suffices to consider admissible extensions.

Skeptical Acceptance

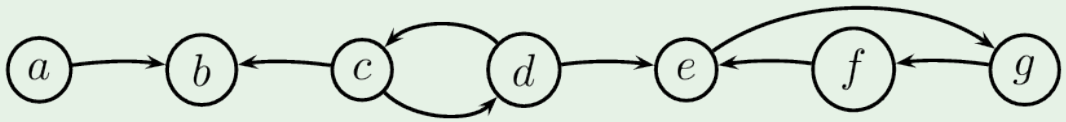
Given an AF $F = (A, R)$ and an argument $x \in A$.
Is x in every preferred extension ?

Complexity:

- The credulous acceptance problem is NP-complete.
- The skeptical acceptance problem is Π_2^p-complete.

Tree-Decomposition

Argumentation Framework

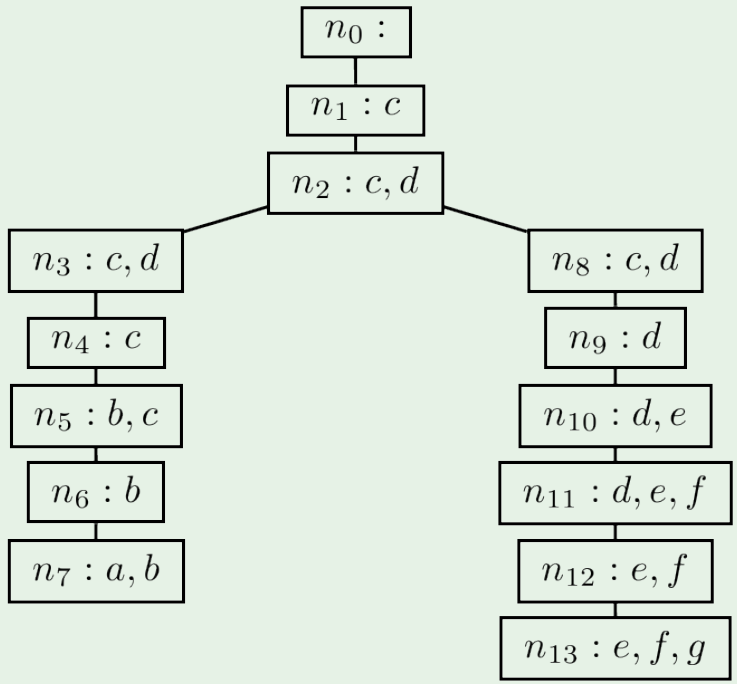


Properties

For an AF $F = (A, R)$:

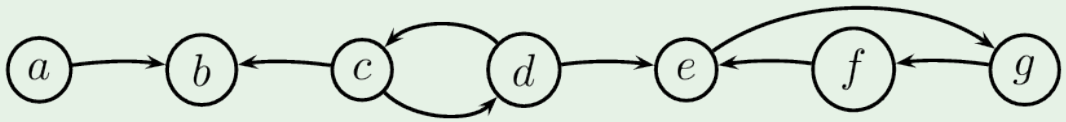
1. Each argument $a \in A$ and each attack $(b, c) \in R$ is contained in at least one bag
2. Bags containing the same argument are connected

Tree-Decomposition



Tree-Decomposition

Argumentation Framework

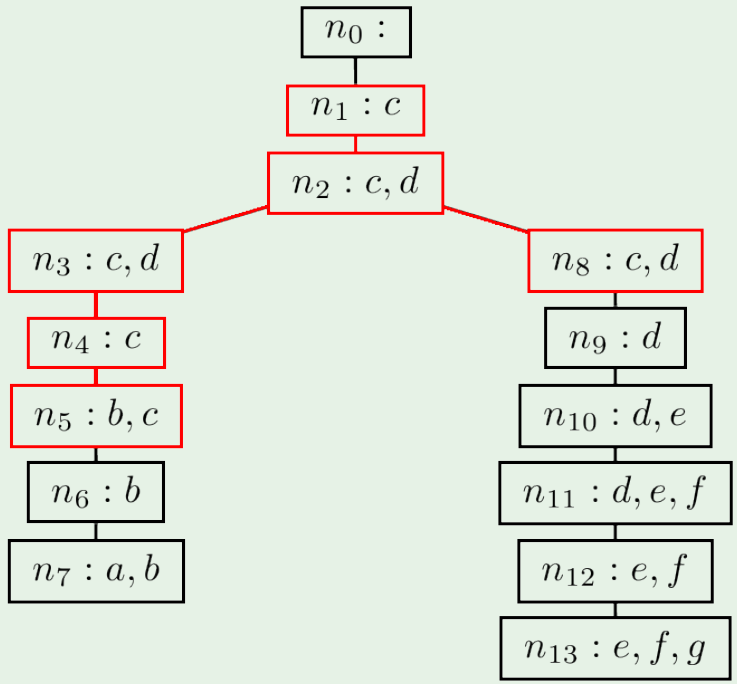


Properties

For an AF $F = (A, R)$:

1. Each argument $a \in A$ and each attack $(b, c) \in R$ is contained in at least one bag
2. Bags containing the same argument are connected

Tree-Decomposition



Tree-Width

Width

The width of a tree-decomposition is the size of the largest bag - 1.

Tree-Width

The tree-width of an AF is the minimum width over all possible tree-decompositions.

Dynamic Programming

Basic Ideas:

- Compute locally admissible / preferred sets for each bag with a bottom-up algorithm on the tree-decomposition
  - For each bag t we only store information about arguments in X_t
  - The information about the "forgotten" arguments is implicitly encoded
- The results for the entire problem can be read from the root

Dynamic Programming

Basic Ideas:

- Compute locally admissible / preferred sets for each bag with a bottom-up algorithm on the tree-decomposition
  - For each bag t we only store information about arguments in X_t
  - The information about the "forgotten" arguments is implicitly encoded
- The results for the entire problem can be read from the root

Theorem

Given an AF $F = (A, R)$ of tree-width w and an argument $a \in A$, the DP-algorithm decides if x is accepted in time $O(f(w) \cdot |F|)$.

Dynamic Programming

Basic Ideas:

- Compute locally admissible / preferred sets for each bag with a bottom-up algorithm on the tree-decomposition
  - For each bag t we only store information about arguments in X_t
  - The information about the "forgotten" arguments is implicitly encoded
- The results for the entire problem can be read from the root

Theorem

Given an AF $F = (A, R)$ of tree-width w and an argument $a \in A$, the DP-algorithm decides if x is accepted in time $O(f(w) \cdot |F|)$.

For details see [Dvořák, Pichler and Woltran KR'10].

Implementation

Our implementation builds on the SHARP[1] framework.

- SHARP offers ready-to-use implementations of various tree decomposition heuristics and
- helper methods for the Preprocessing and Dynamic Algorithm steps.
- It provides normalized tree decompositions, i.e. there are just four specific types of nodes.
- To implement our algorithms we just have to provide the methods and data structures for these four node types.

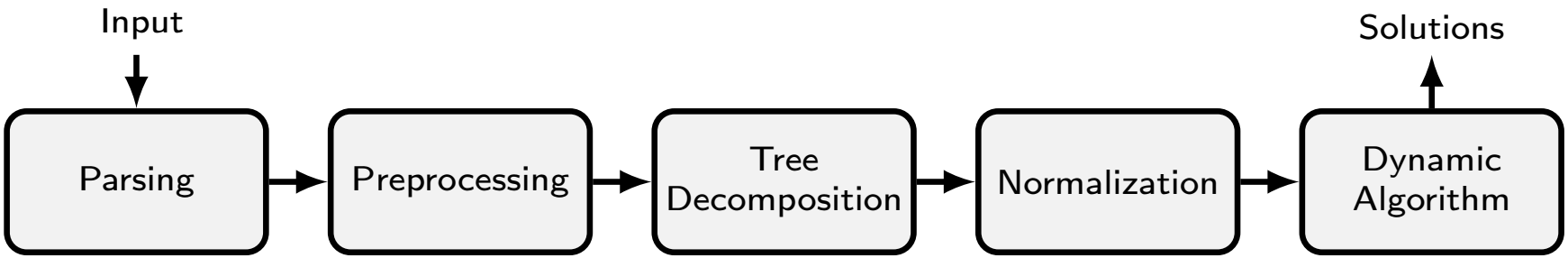


Figure: Architecture of the *SHARP* framework.

[1] http://www.dbai.tuwien.ac.at/research/project/sharp/

Preliminary Experiments

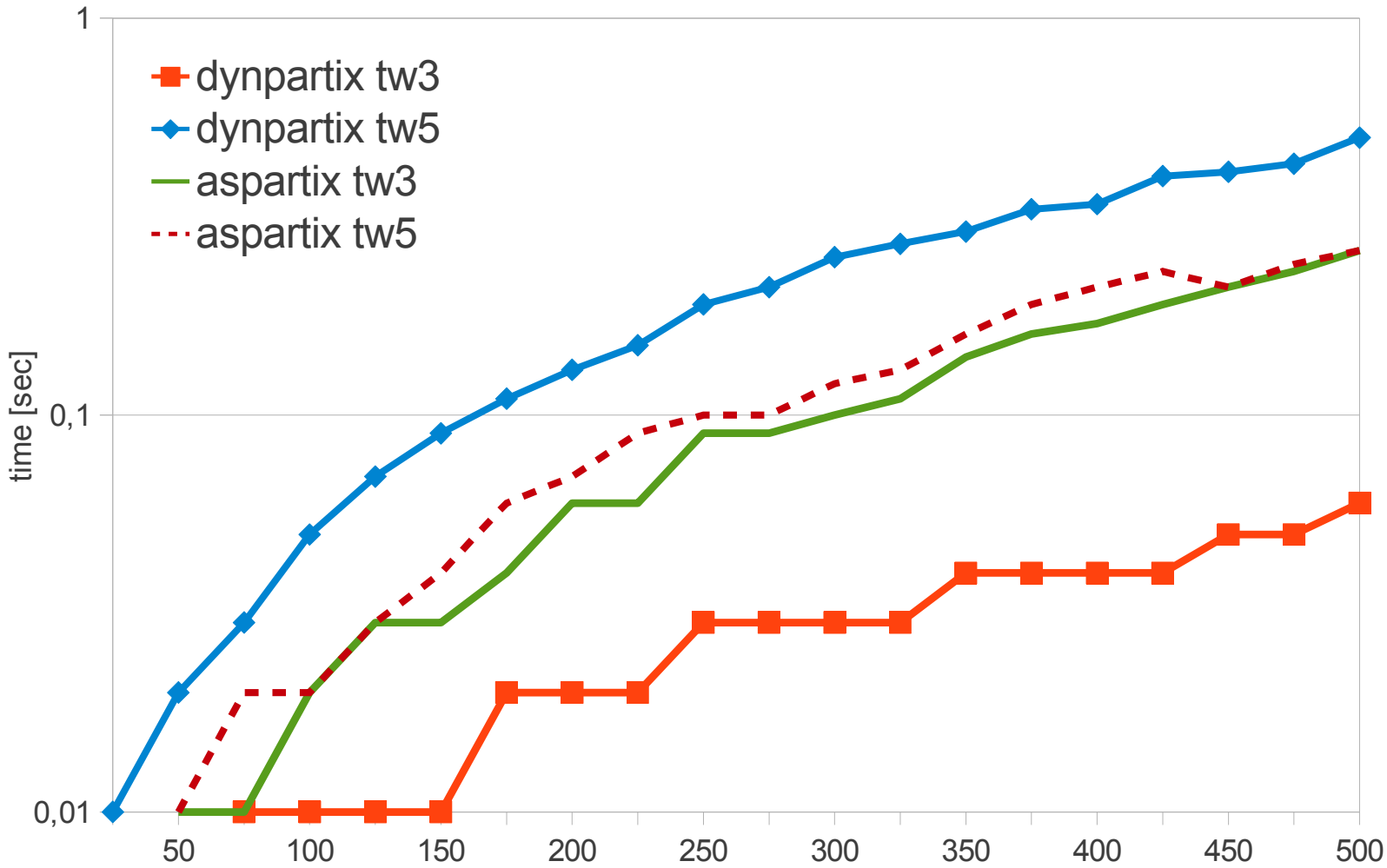
We compare dynPARTIX with ASPARTIX[2], one of the most efficient tools for abstract argumentation, on instances of low tree-width.

Test system:
Intel®CoreTM 2 CPU 6300@1.86GHz
SUSE Linux version 2.6.27.48

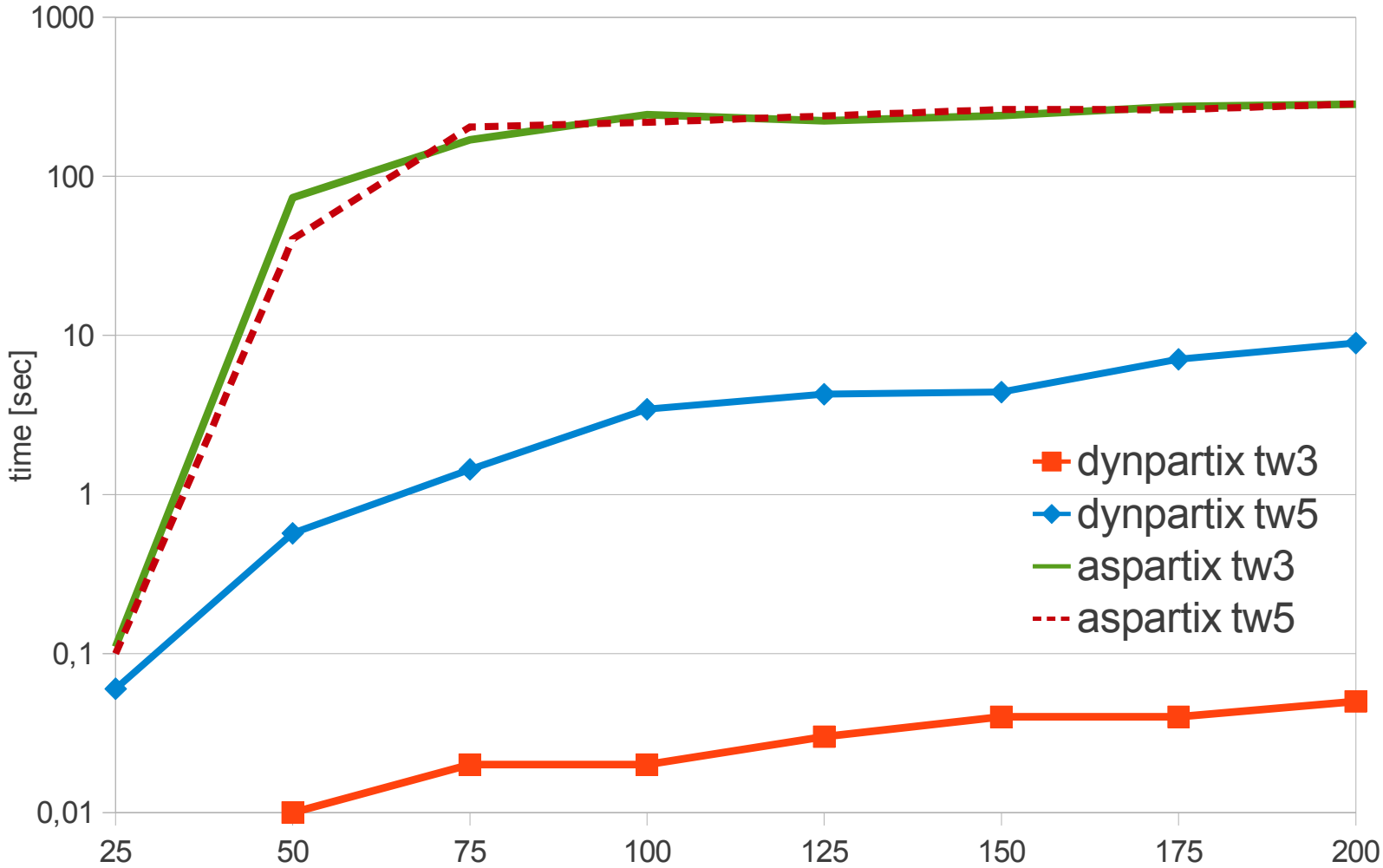
Test instances:
We randomly generated 4800 AFs varying the number of arguments; the tree-width; and the number of attacks. (To ensure that AFs are of a certain tree-width we used random grid-structured AFs.)

[2] www.dbai.tuwien.ac.at/research/project/argumentation/systempage/

Experimental Results for Credulous Acceptance



Experimental Results for Skeptical Acceptance



Conclusion

- We presented dynPARTIX, a implementation of fixed-parameter tractable algorithms w.r.t. tree-width for
  - Credulous Reasoning
  - Skeptical Reasoning
  - Counting Extensions
  - Enumerating Extensions
- The techniques presented for preferred semantics are prototypical, i.e. can be easily applied to several other semantics.
- We provided preliminary experiments that underpin that our DP-algorithms pay off on instances of low tree-width.

Conclusion

- We presented dynPARTIX, a implementation of fixed-parameter tractable algorithms w.r.t. tree-width for
  - Credulous Reasoning
  - Skeptical Reasoning
  - Counting Extensions
  - Enumerating Extensions
- The techniques presented for preferred semantics are prototypical, i.e. can be easily applied to several other semantics.
- We provided preliminary experiments that underpin that our DP-algorithms pay off on instances of low tree-width.

http://www.dbai.tuwien.ac.at/research/project/argumentation/dynpartix/