Overview
- Argumentation Frameworks (AFs) important research field in Artificial Intelligence
- Selection of 'appropriate' arguments from AF defined by a semantics
- Many different semantics
- Selection oftentimes computationally hard (intractable)
- Identify tractable fragments
  - For AFs: Tree-Width, defined on Tree Decompositions

Goal
- Development of novel algorithms for stable, complete and admissible semantics, following up the work of Dvořák et.al. [1]
- Based on Tree Decompositions, tractable
- Evaluation of different types of Tree Decompositions

Argumentation Framework [2]
Pair \( F = (A, R) \) where \( A \) is a set of arguments and \( R \subset A \times A \) is the attack relation.

Tree Decomposition [3]
Tree where each node has a bag that contains a set of vertices from the original graph such that:
- every vertex in at least one bag
- connected vertices together in bag
- nodes containing a vertex are connected upwards the tree.

Admissible Semantics [2]
Set \( S \) admissible if conflict-free and each argument in \( S \) is defended.

Stable Semantics [2]
Set \( S \) stable if conflict-free and each argument not in \( S \) is attacked by \( S \).

Complete Semantics [2]
Set \( S \) complete if admissible and each defended argument is in \( S \).

Motivation

Basic Definitions

Approach for Algorithms based on Tree Decompositions

Preparation

Computation

V-Colorings: Defined solely on current vertices and colorings of vertices in the sub-tree

- Restricted Sets: Contain arguments that that were completely considered in the sub-tree of a node, arguments fulfill properties of semantics
- Colorings: Defined on restricted sets
- Encode information about relations by assigning colors to each argument of current node

Result Delivery

- At root node all arguments have been handled
- Restricted sets correspond to extensions of AF
- Fixed-parameter tractability achieved

Test Setup

- Comparison of already existing algorithms for admissible semantics on normalized tree decompositions [1] to novel on semi-normalization
- Different test instance types (Grid, Clique)
- Different width and edge probability

Analysis of Benchmarks

- Semi-normalized implementation outperforms normalized in every test case
- Relative performance gain significant (up to 50%)
- Absolute performance gain depends on cost for preparation and branch node evaluation
- In general: Less edges, performance gain more significant

Experimental Results

Conclusion

Contributions
- Novel algorithms for stable and complete semantics based on normalized tree decompositions
- Novel algorithm for admissible semantics based on semi-normalized tree decompositions
- Implementations and correctness proofs of the algorithms
- Experimental results show that algorithm on semi-normalized tree decompositions outperforms the existing one

Future Work
- Provide algorithms for further semantics based on tree decompositions
- Analysis of run-time on non-normalized tree decompositions
- Detailed complexity analysis

Diplomarbeit: Datenbanken und Künstliche Intelligenz

E-Mail: gcharwat@dbai.tuwien.ac.at

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