Defining Argument Weighing Functions

Thomas F. Gordon
Fraunhofer FOKUS
Berlin, Germany

November 2016

Outline

- Weighing and Balancing Arguments
- Limitations of Dung Abstract Argumentation Frameworks
- Summary of Our Framework for Weighing and Evaluating Arguments
- Example Weighing Functions
- Related Work
- Conclusions

Weighing and Balancing Arguments (Application Scenarios)

- Practical reasoning. Balancing pros and cons of alternative actions
- Theoretical argumentation. Constructing and comparing alternative theories. Balancing multiple criteria to choose the most coherent theory.
- Factual argumentation. Balancing conflicting evidence (e.g. testimony). Constructing and comparing alternative narratives (“stories”). Balancing multiple criteria to choose the most coherent narrative.
- Arguing about open-textured concepts (subsumption). Balancing different methods of interpretation (e.g. literal, historical, teleological). Balancing interests to preserve “proportionality”.

Dung Abstract Argumentation Frameworks (1995)

AF = (Arguments, Attacks)

Not intended to handle balancing, but rather only to resolve attack relations among arguments:

“The goal of this paper is to give a scientific account of the basic principal 'The one who has the last word laughs best' of argumentation …”

Dung's Argumentation Evaluation Pipeline

Argument Weights

- The weight of an argument can depend on the labels (in, out, undecided) of its premises
- The failure of a premise can decrease or increase (!) the weight of the argument
- Examples:
  - Corroborative testimony. An argument from witness testimony is strengthened if there are further witnesses.
  - A fortiori argument: An argument for some option is strengthened if the option is even better than claimed in the argument. (That is, if the claim is false!)

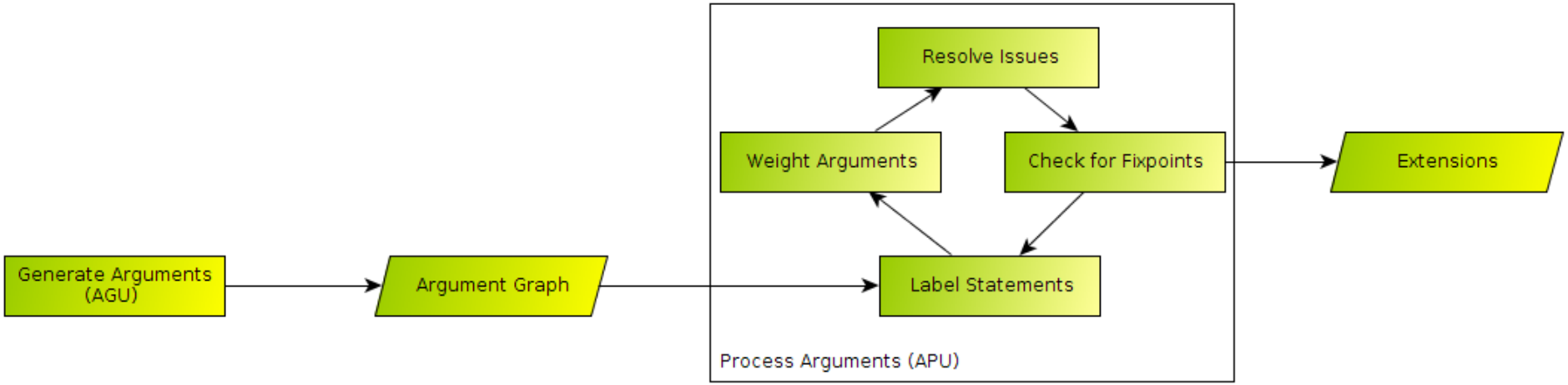
Problem with Dung's Pipeline When Balancing Arguments

- Argument weights and statement labels can be recursively dependent on each other:
  - The weight of an argument can depend on the label of the statements which are its premises, and **recursively**
  - The label of a statement can depend on the weights of the arguments pro and con this statement.
- Thus, the weights of arguments cannot always be computed before the labels of statements, as in Dung’s pipeline model.

Our Formal Model of Balancing Arguments

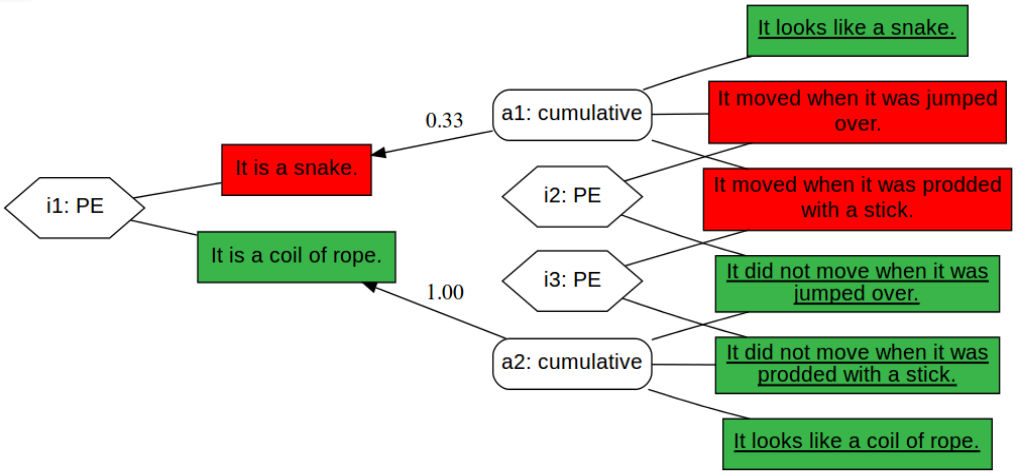
- Gordon, Thomas F. and Walton, Douglas (2016). Formalizing Balancing Arguments. Proceedings of the 2016 conference on Computational Models of Argument (COMMA 2016) (pp. 327-338)

Recursive Process Model



Model: Argument Graphs

- Tripartite directed graphs (may include cycles)
- Node types
  - Statements
  - Arguments
  - Issues
- Statements are premises and conclusions of arguments and options of issues



Example Argument

- Id: a1
- Scheme: car-buying-scheme
- Premises
  - type(porsche,sports)
  - price(porsche,high)
  - safety(porsche,medium)
  - speed(porsche,fast)
- Conclusion
  - buy(porsche)
- Undercutter: ¬app(a1)

Iterative Evaluation Procedure (Basic Idea)

- Evaluation
  - Statements are labeled in or out. Initially undecided.
  - Arguments are weighed (0.0 to 1.0). Initially no value (nil).
- Statements are labeled as much as possible on each iteration, starting with assumptions
- Arguments are weighed, by applying weighing functions, after their premises have been labeled
- Issues are resolved, using proof standards, when all the arguments for each option have been weighed
- As with Dung AFs, different fixpoint semantics are possible (grounded, preferred, etc)

Argument Weighing Functions

- The model is a framework, instantiated by
  - A language (*L*, a finite set of statements)
  - Argumentation schemes with weighing functions
  - Proof standards (e.g. preponderance of the evidence)
- Notice that argumentation schemes have been extended with weighing functions in this model.

Signature of Argument Weighing Functions

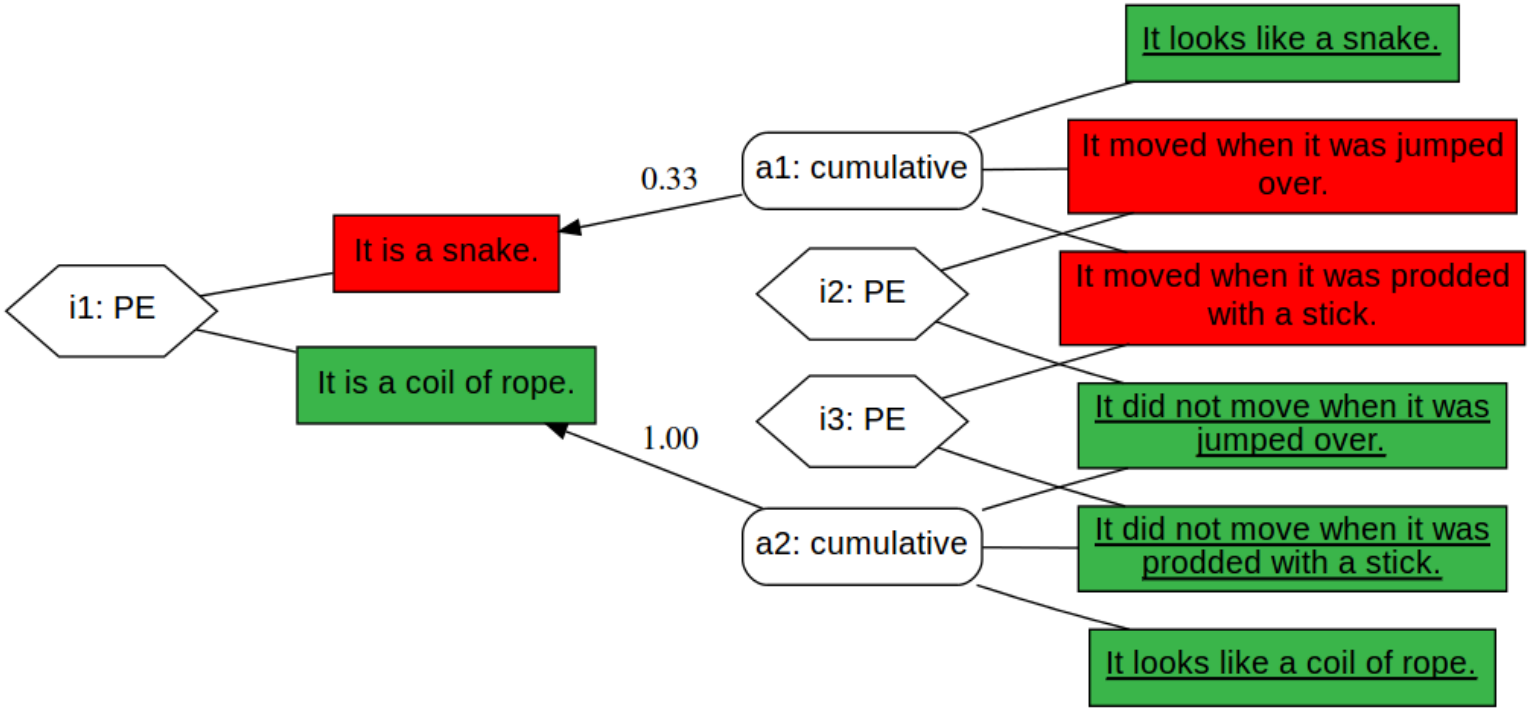
labeling x argument graph x argument → [0..1.0]

- where a labeling is a mapping from statements in *L* to {in, out undecided} , and the argument is the one being weighed.
- Notice that the weight of an argument may depend on the labels of any statements in the argument graph, not just its own premises.

Simple Examples of Weighing Functions

- Linked Argument
  - 1.0 if all premises are in
  - 0.0 otherwise
- Convergent Argument
  - 1.0 if some premise is in
  - 0.0 otherwise
- Cumulative Argument
  - number of in premises / total number of premises
- Factorized Argument
  - number of in factors / total number of factors

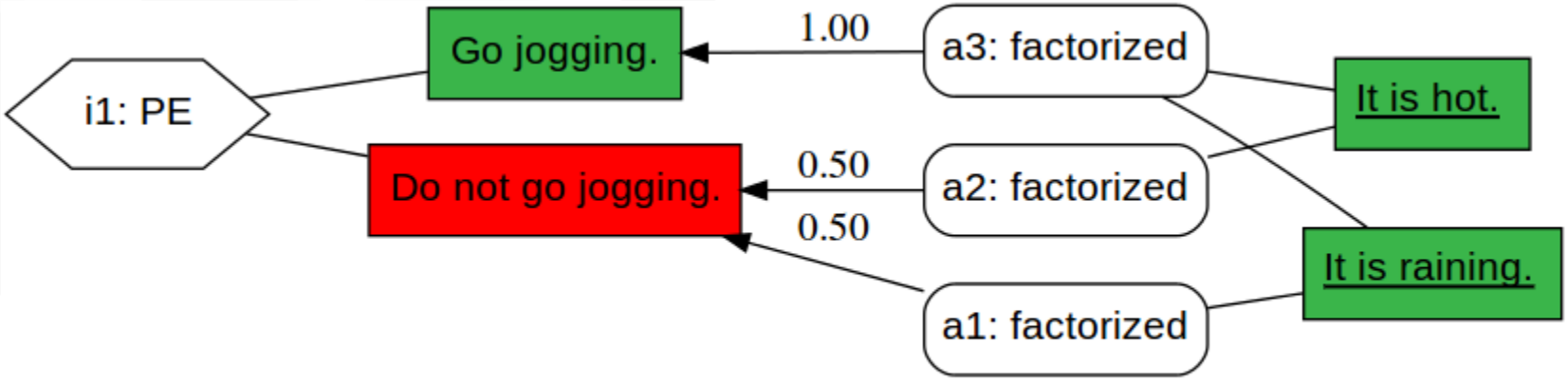
Example Cumulative Argument: Snake or Rope?



Weight = number of in premises / total number of premises

Walton, D. N.; Tindale, C. W. & Gordon, T. F. (2014), 'Applying Recent Argumentation Methods to Some Ancient Examples of Plausible Reasoning', Argumentation 28(1), 85–119. (Doi: 10.1007/s10503-013-9306-y.)

Example of Factorized Arguments



Weight = number of in factors / total number of factors

Prakken, H. A study of accrual of arguments, with applications to evidential reasoning, In Proceedings of the 10th International Conference on Artificial Intelligence and Law, ACM, (2005), 85-94.

Implementation

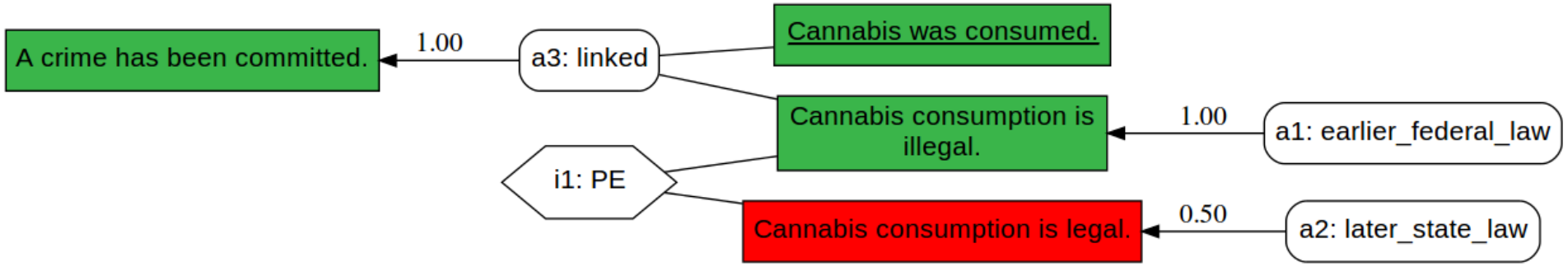
- Carneades, Version 4
- Online at http://carneades.fokus.fraunhofer.de/carneades/
- Source code at https://github.com/carneades/carneades-4
- Includes
  - a language for defining argumentation schemes and some kinds of weighing functions
  - an inference engine, based on Constraint Handling Rules (CHR) for generating arguments
  - an argument graph evaluator
  - an argument mapping (visualization) tool

Example: lex.yml
Weighing Arguments by Sorting Their Schemes

```
weighing_functions:
    lex:  # lex superior followed by lex posterior
        preference:
          - property: authority
            order: [local, state, federal] # weakest to strongest
          - property: effective_date
            order: ascending   # later dates have higher priority
```

```
argument_schemes:
    - id: earlier_federal_law  # some federal law
      meta:
        authority: federal
        effective_date: 1989-06-14
      weight: lex
      … # premises and conclusions omitted
    - id: later_state_law  # some state law
      meta:
        authority: state
        effective_date: 2008-04-13
      weight: lex
      ...
```

- Arguments are weighed here by sorting the *schemes* used to construct the arguments, using properties of the schemes.
- Weights are automatically assigned in a way which respects the ordering of the schemes
- The two schemes in this example share the same weighing function, named “lex”

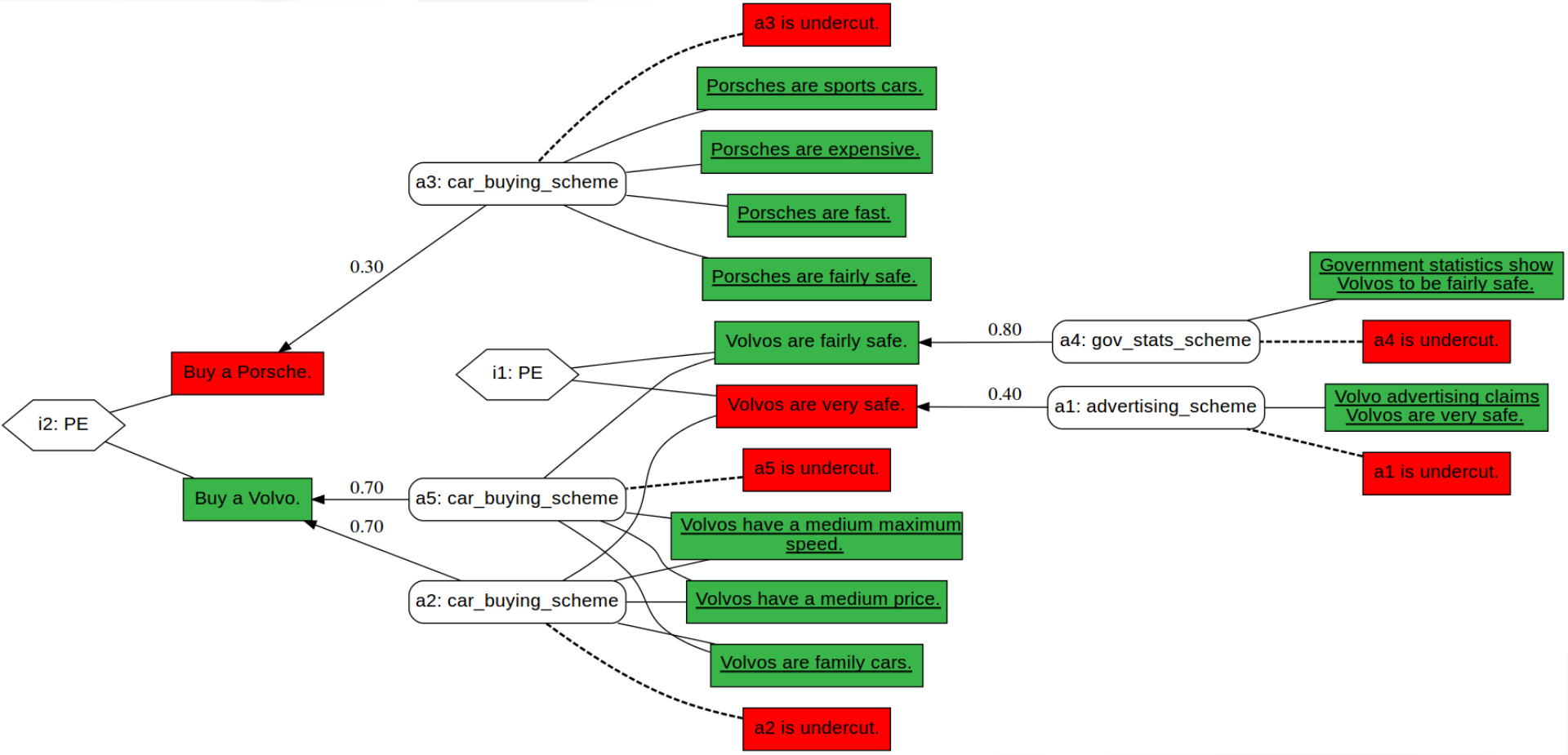

A crime has been committed.
1.00
a3: linked
Cannabis was consumed.
Cannabis consumption is illegal.
1.00
a1: earlier_federal_law
i1: PE
Cannabis consumption is legal.
0.50
a2: later_state_law

Multiple-Criteria Decision Analysis (MCDA)

```
id: car_buying_scheme
 variables: [C,P,S,F,T]
 weight:
   criteria:
     hard: []  #  premises which must be in, none here
     soft:       # soft constraints
       price:
         factor: 2
         values: {low: 1.0, medium: 0.5, high: 0.0}
       type:
         factor: 2
         values: {sports: 0.0, sedan: 0.5, family: 1.0}
       speed:
         factor: 2
         values: {slow: 0.0, medium: 1.0, fast: 0.5}
       safety:
         factor: 4
         values: {low: 0.0, medium: 0.5, high: 1.0}
premises:
   - price(C,P)
   - type(C,T)
   - speed(C,S)
   - safety(C,F)
conclusions:
   - buy(C)
```

Idea: the weight of the argument is the **weighted sum** of the **proven properties** of a given option

Example: How to Buy a Porsche



Brewka, Gerhard, and Thomas F. Gordon. "How to Buy a Porsche: An Approach to Defeasible Decision Making." Working Notes of the AAAI-94 Workshop on Computational Dialectics. Seattle, Washington, 1994. 28-38.

Some Related Work

- ASPIC+
- Abstract Dialectical Frameworks (ADF)

ASPIC+

- Prakken, Henry (2010). An abstract framework for argumentation with structured arguments. Argument & Computation, 1, 93-124.
- “Weighs” arguments based on a static ordering of the rules in a knowledge base used to construct the arguments.
- Cumulative arguments (accrual) can be simulated by creating additional arguments for each subset of the premises. But this causes an exponential blow-up in the number of arguments.
- Is based on Dung AFs and thus is limited by its pipeline model of argument evaluation.

Abstract Dialectical Frameworks (ADFs)

- Brewka, Gerhard and Woltran, Stefan (2010). Abstract Dialectical Frameworks. Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (pp. 102-111), AAAI Press.
- Convenient generalization of Dung AFs for defining a wide-variety of graph-based formalisms.
- But labels of nodes can depend only on their parent nodes.
- This is not general enough to weight arguments using multi-criteria decision analysis, where the weights of arguments depend on labels of statements further away in the graph.

Conclusions

- Original formal model of structured argument providing a framework for defining and applying a wide-range of argument weighing functions
- Illustrated with a several examples, including
  - Sorting arguments by their meta-level properties. Legal example, with lex posterior and lex superior.
  - Cumulative arguments (accrual). Snake and rope example.
  - Factorized arguments. Jogging example.
  - Multiple-criteria decision analysis. Car buying example
- Fully implemented, in Carneades 4
- Caveat: Does not constrain weighing functions to only sensible, useful or meaningful ones!

Thank You!

Acknowledgements

- Carneades project. Canadian Social Sciences and Humanities Research Council
- European Policy Compass Project (https://policycompass.eu)

Policy Compass

Open data-driven analysis and impact evaluation