

Virtual Arguments

On the Design of Argument Assistants for Lawyers and Other Arguers

Bart Verheij
Artificial Intelligence, University of Groningen, The Netherlands

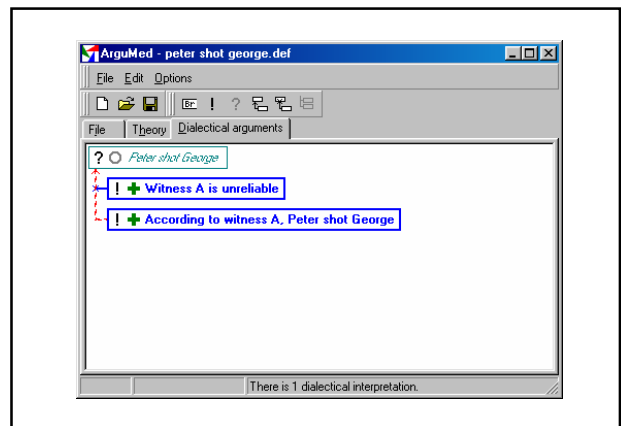
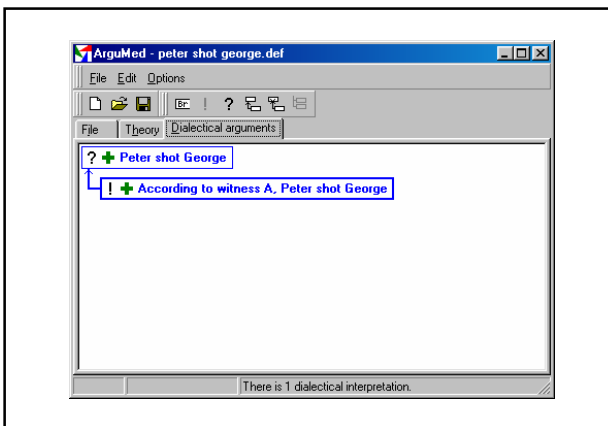


Argumentation software

Argument assistants are computer programs that support argumentative tasks

Analogy:

Text writing assistants (aka word processing software) are computer programs that support text writing tasks

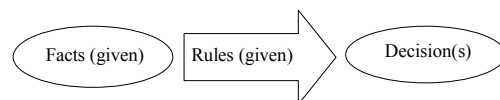


ArguMed

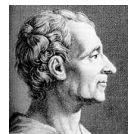
- Warranted defeasible argumentation
- Argument construction, argument drafting
- Natural interaction, 'natural argumentation'
- Argument evaluation
- Underlying logic of defeasible argumentation

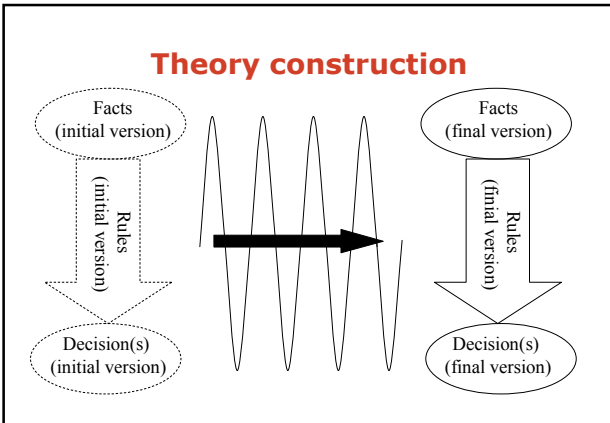
Verheij, B. (2005). *Virtual Arguments. On the Design of Argument Assistants for Lawyers and Other Arguers*. T.M.C. Asser Press, The Hague.
Verheij, Bart (2003). Artificial argument assistants for defeasible argumentation. *Artificial Intelligence*, Vol. 150, No. 1-2, pp. 291-324.
Verheij, Bart (2003). DefLog: on the logical interpretation of prima facie justified assumptions. *Journal of Logic and Computation*, Vol. 13, No. 3, pp. 319-346.

The subsumption model

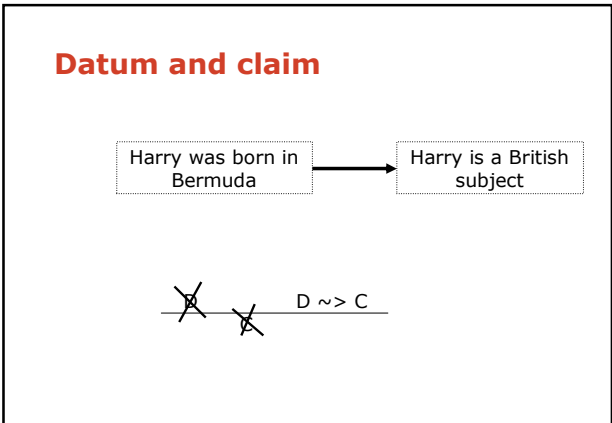
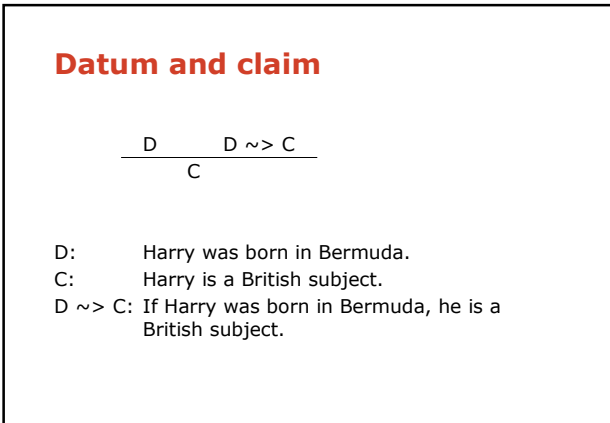
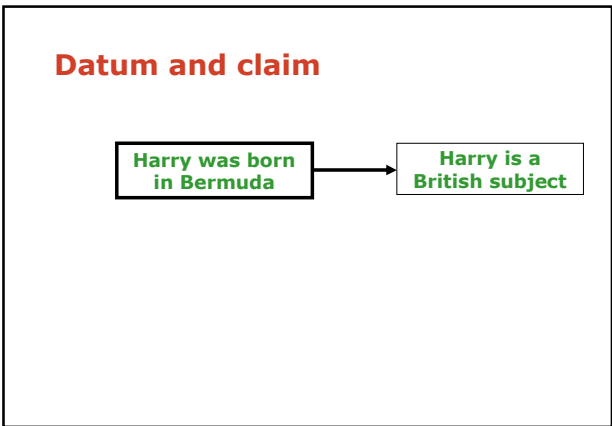
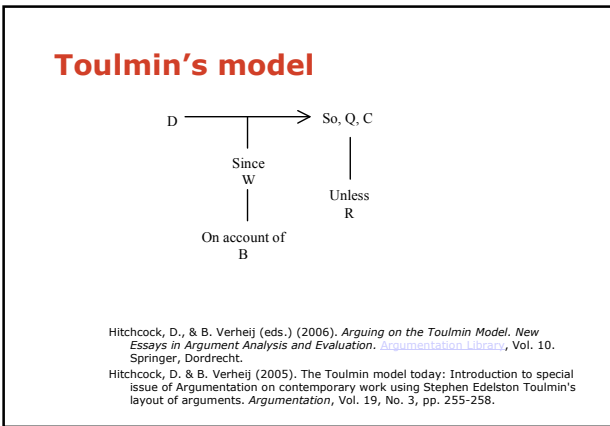


Montesquieu (1689-1755):
The judge as 'bouche de la loi'

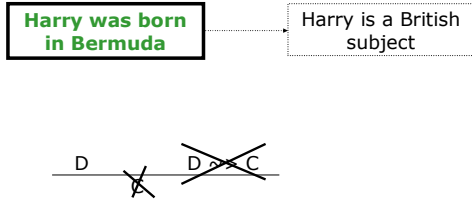




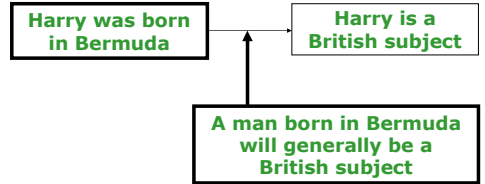
1. Introduction
 - Argumentation software
 - Theory construction
2. **ArguMed & DefLog**
3. Beyond boxes and arrows
4. Challenges



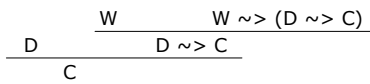
Datum and claim



Warrant



Warrant



- D: Harry was born in Bermuda.
- C: Harry is a British subject.
- W: A man born in Bermuda will generally be a British subject.

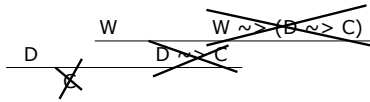
Warrant

A man born in Bermuda will generally be a British subject.

If *Person* was born in Bermuda, then generally *Person* is a British subject.

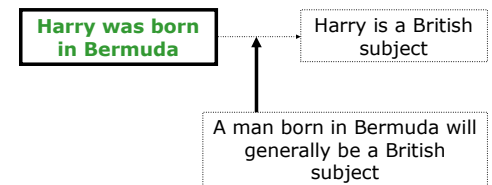
If Harry was born in Bermuda, then generally he is a British subject.

Warrant

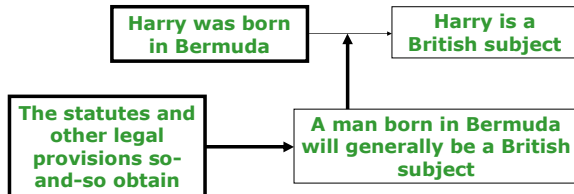


- D: Harry was born in Bermuda.
- C: Harry is a British subject.
- W: A man born in Bermuda will generally be a British subject.

Warrant



Backing



Backing

$$\frac{B \quad B \sim > W}{W} \quad W \sim > (D \sim > C)$$

$$\frac{D}{C} \quad D \sim > C$$

- B: The statutes and other legal provisions so-and-so obtain.
- W: A man born in Bermuda will generally be a British subject.
- $B \sim > W$: If the statutes and other legal provisions so-and-so obtain, a man born in Bermuda will generally be a British subject.

1. Introduction

Argumentation software
Theory construction

2. ArguMed & DefLog

Datum & claim
Warrant & backing
Rebuttal

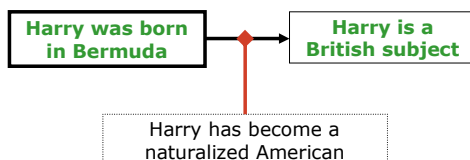
3. Beyond boxes and arrows

4. Challenges

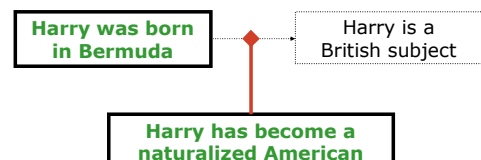
Rebuttal I (no warrants)



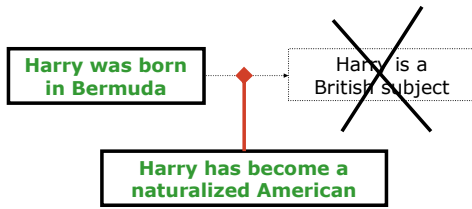
Rebuttal I (no warrants)



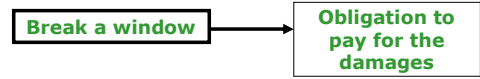
Rebuttal I (no warrants)



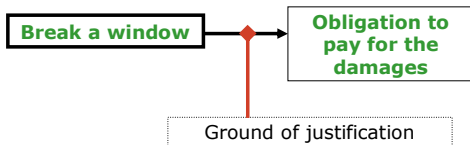
Rebuttal I (no warrants)



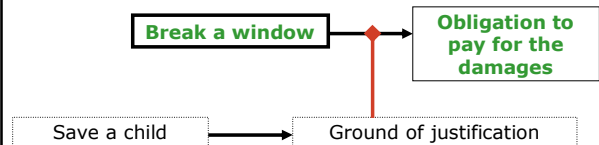
Reinstatement



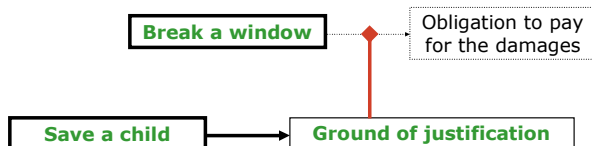
Reinstatement



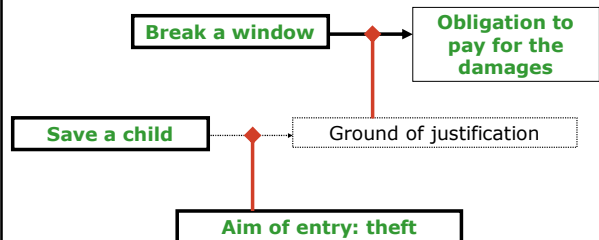
Reinstatement

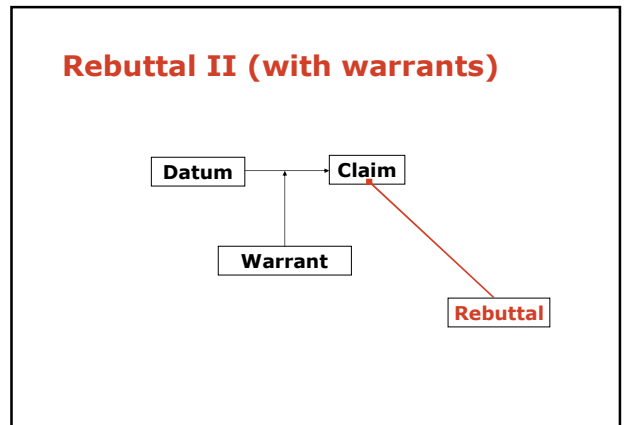
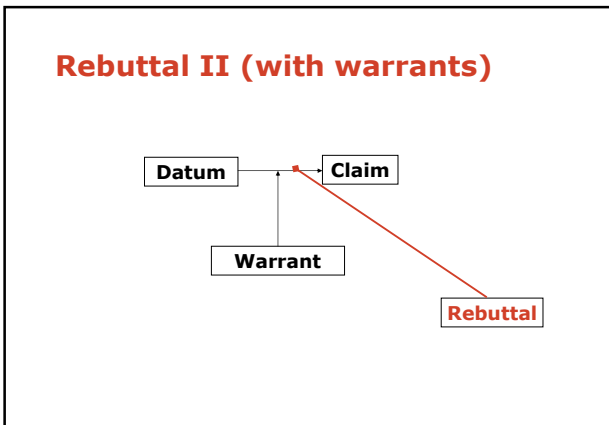
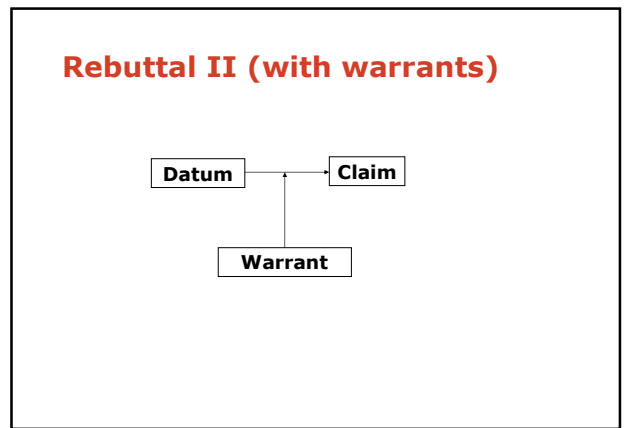
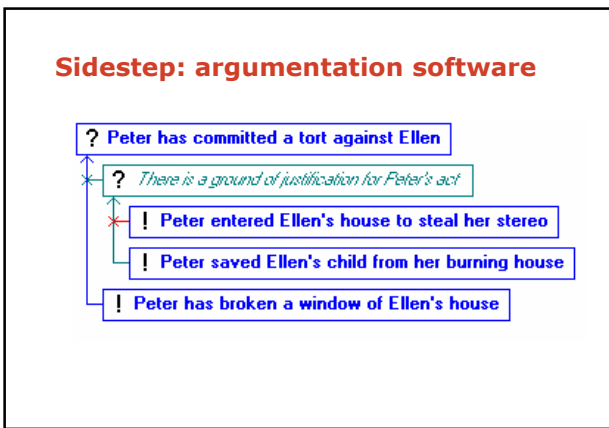
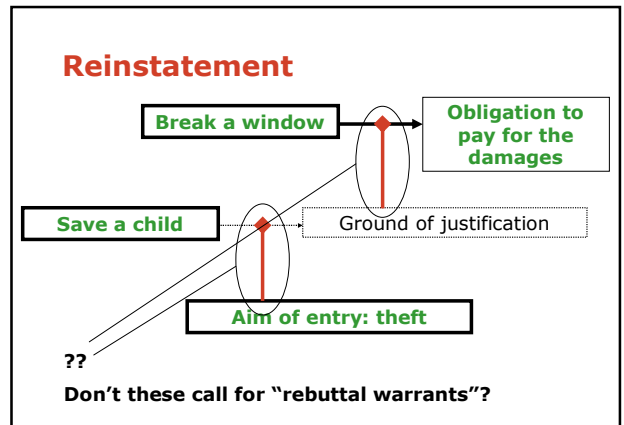
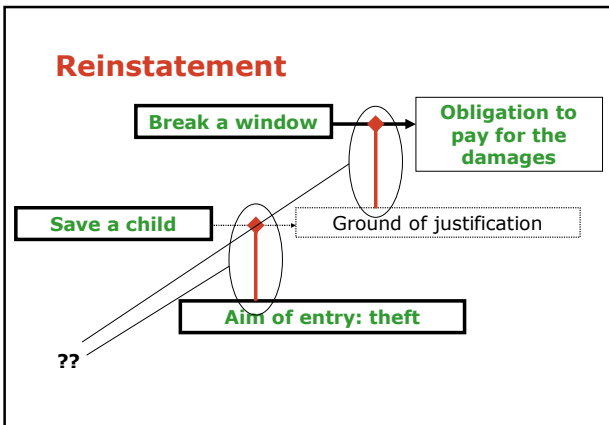


Reinstatement

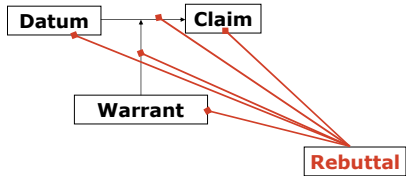


Reinstatement





Rebuttal II (with warrants)

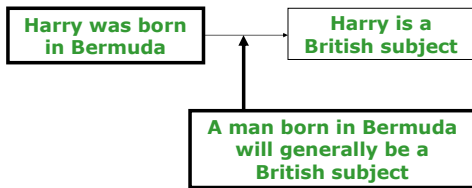


Rebuttal II (with warrants)

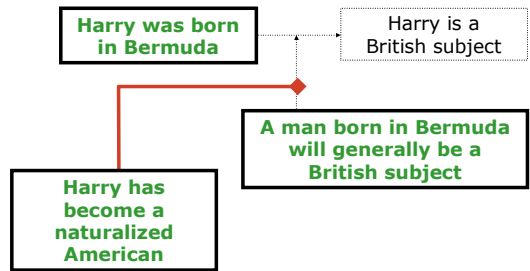
If we look at the warrant-datum-claim part of Toulmin's scheme, there are five statements that can be argued against:

- The datum *D*
- The claim *C*
- The warrant *W*
- The implicit conditional 'If *D*, then *C*' that expresses the bridge from datum to claim.
- The implicit conditional 'If *W*, then if *D*, then *C*' that expresses the bridge between warrant and the previous implicit conditional.

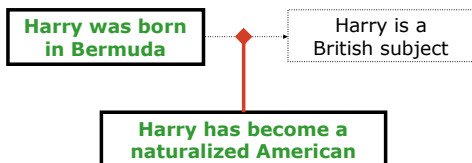
Rebuttal II (with warrants)



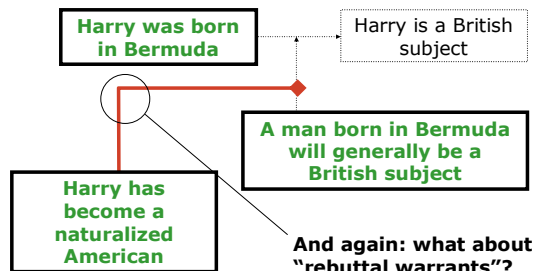
Rebuttal II (with warrants)



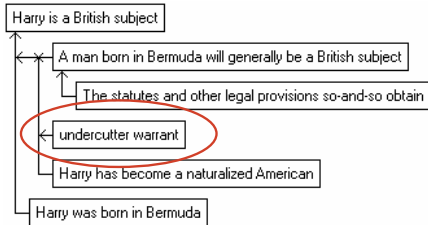
Rebuttal I (no warrants)



Rebuttal II (with warrants)



Sidestep: argumentation software



1. Introduction

Argumentation software
Theory construction

2. ArguMed & DefLog

Datum & claim
Warrant & backing
Rebuttal I (no warrants)
Reinstatement
Rebuttal II (with warrants)
DefLog & argumentation formalisms

3. Beyond boxes and arrows

4. Challenges

The dialectical logic DefLog

A conditional $\sim >$ that validates Modus ponens
A connective \times that expresses 'dialectical negation'

Theories are interpreted as sets of defeasible assumptions

pro: $\varphi \sim > \psi$
con: $\varphi \sim > \times \psi$
warrant: $\varphi \sim > (\psi \sim > \chi)$
undercutter: $\varphi \sim > \times (\psi \sim > \chi)$
rebutter: $((\varphi \sim > \psi) \wedge \varphi) \sim > \times (\chi \sim > \text{not-}\psi)$
or $\psi \sim > \times (\chi \sim > \text{not-}\psi)$

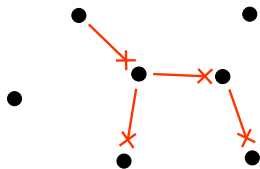
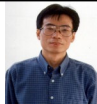
Pollock's undercutting and rebutting defeaters



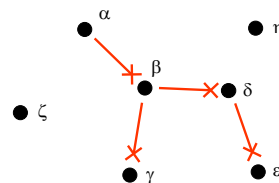
If P is a prima facie reason for Q, then R is a *rebutting defeater* if and only if R is a reason for denying Q.

Undercutting defeaters attack the connection between the reason and the conclusion rather than attacking the conclusion itself.

Dung (1995): argument attack as a directed graph



Admissible sets



Admissible, e.g.: $\{\alpha, \gamma\}$, $\{\alpha, \gamma, \delta, \zeta, \eta\}$
Not admissible, e.g.: $\{\alpha, \beta\}$, $\{\gamma\}$

1. Introduction

Argumentation software
Theory construction

2. ArguMed & DefLog

Datum & claim
Warrant & backing
Rebuttal I (no warrants)
Reinstatement
Rebuttal II (with warrants)
DefLog & argumentation formalisms

3. Beyond boxes and arrows

4. Challenges

Reason-Based Logic



Key elements of Reason-Based Logic are the formal modeling of:

Rules as objects
The exclusion of rules
The weighing of reasons
Rule application

Types of facts

Valid(rule(condition, conclusion))

Reason(fact, state_of_affairs)

Excluded(rule(condition, conclusion))

Outweighs(reasons_pro, reasons_con, state_of_affairs)

(1) *P*. If *P* then *Q*.
Therefore *Q*.

(2) All *P*s are *Q*s. Some *R* is not a *Q*.
Therefore some *R* is not a *P*.

(3) Person *E* says that *P*. Person *E* is an expert with respect to the fact that *P*.
Therefore *P*.

(4) Doing act *A* contributes to goal *G*. Person *P* has goal *G*.
Therefore person *P* should do act *A*.

Argumentation schemes

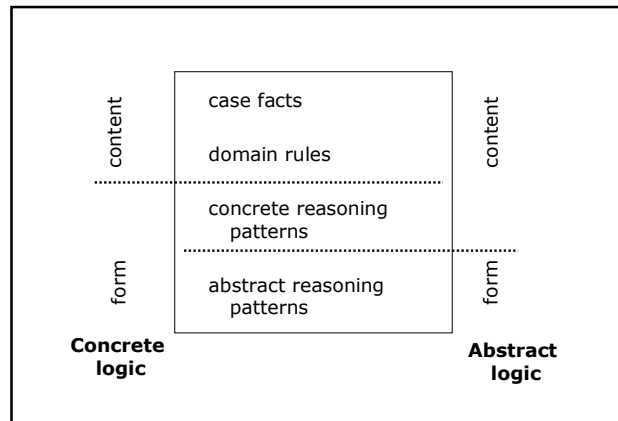


(3) Person *E* says that *P*. Person *E* is an expert with respect to the fact that *P*.
Therefore *P*.

(4) Doing act *A* contributes to goal *G*. Person *P* has goal *G*.
Therefore person *P* should do act *A*.

A method for the investigation of argumentation schemes

1. Determine the relevant types of sentences
2. Determine the argumentation schemes
3. Determine the arguments against the use of the argumentation schemes
4. Determine the conditions for the use of the argumentation schemes



Domain rules

- (5) Person *P* has committed crime *C*. Crime *C* is punishable by *n* years of imprisonment. Therefore person *P* can be punished with up to *n* years of imprisonment.
- (6) Person *A* is an unmarried man. Therefore person *A* is a bachelor.

Legal skills: written case solutions

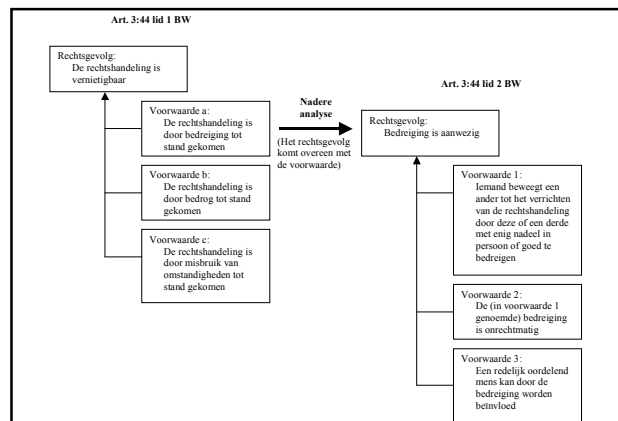
Task:

- solve legal cases, i.e., determine the legal consequences of cases, and
- write an argumentative text in a structured and legally valid way



Legal skills: written case solutions

- Provide examples of written case solutions, instead of a theoretical framework
Good examples of written case solutions
- Focus on examples with meaningful content, not on formal ones
No Modus ponens, perhaps not even affirming the consequent
- Train an analytic attitude, not a particular formal system
Let students actually write case solutions; grade by quality of solution, not by adherence to a system
- Show content-based difficulties in examples
Legal reasoning is hard because of its content, (almost) never because of its logical structure.



The Argument Machine

The following might be more true than we want:

A legal professional isn't really helped much with visualizations.

He needs check-lists.

He needs memory extensions.

He needs integrated word, content & argument processing.



The screenshot shows a legal research interface. On the left, there is a list of legal topics with checkboxes and search icons. The topics include: Onrechtmatigheid, Inbreuk op een recht, Strijd met wettelijke plaats, Strijd met ongeschreven recht, Rechtsvaardigheidsgrond, Toerekenbaarheid, Schade, Toerekenbaarheid volgens wet, Toerekenbaarheid volgens ongeschreven recht, Levensduur langer dan 14 jaar, Geestelijke of lichamelijke toestand, Het beseffen van daad, Schade, Causaal verband, and Resultaatsverste. On the right, there is a detailed view of a specific case, including the case number (Stad. 41.1992.621), the theme (Thema 'Strijd met ongeschreven recht'. Ongeval bij tennispartij.), and the subject (Stad. 41.1998.192). The interface also includes a search bar at the top right and a 'Opslaan' button at the bottom.

1. Introduction

Argumentation software
Theory construction

2. ArguMed & DefLog

Datum & claim
Warrant & backing
Rebuttal I (no warrants)
Reinstatement
Rebuttal II (with warrants)
DefLog & argumentation formalisms

3. Beyond boxes and arrows

Reason-Based Logic
Argumentation schemes
Legal skills: written case solutions
The Argument Machine

4. Challenges

Argumentation software: challenges

Natural design

Find natural designs for argumentation software
Prove the naturality of these designs

Usefulness

Find niches of usefulness for argumentation software
Prove the usefulness for these niches

Content

Find ways of providing content in argumentation software
Prove the effectiveness of these ways