Chapter 6

Analyzing argumentation models using CumulA

After the description of the argumentation model CumulA in chapter 5, we show how CumulA can be used to analyze existing argumentation models. We start with a discussion of distinctions that can be made between argumentation models. We make these distinctions precise by showing their formal counterparts for CumulA's argumentation theories. After capturing elements of a number of existing argumentation models in CumulA's argumentation theories, we apply the distinctions to these argumentation theories.

In section 1, we discuss types of arguments. In section 2, we treat argument structure and defeat. We distinguish sentence-type, step-type and composite-type defeat. In section 3, we consider individual and groupwise defeat. In section 4, we characterize triggers of defeat. We distinguish inconsistency-triggered and counterargument-triggered defeat. In section 5, we deal with directions of argumentation. We distinguish forward, backward and bidirectional argumentation. In section 6, we capture elements of several major argumentation models in CumulA's argumentation theories.[1] In section 7, the distinctions made are applied to these argumentation theories. In this way, the argumentation theories capturing elements of existing argumentation models can be compared on formal grounds.

1 Types of arguments

Several types of arguments, that have been proposed in argumentation models, can in CumulA (chapter 5) be distinguished by their structure.

The first type of arguments are the *statements*, that have trivial structure:

Statement.

---

[1] We stress that we give no formal relations between the argumentation models and CumulA's argumentation theories.

Many argumentation models do not deal with structured arguments. For instance, Poole's Logical Framework for Default Reasoning (Poole, 1988)[2] uses special sets of sentences without structure. In Dung's Argumentation Frameworks (Dung, 1993, 1995), arguments are structureless objects, that can attack each other.

The second type of arguments are the *single-step arguments*, which have the simplest non-trivial structure:

Reason.
So, *Conclusion.*

For instance, in Propositional and First-Order Predicate Logic,[3] the semantical and proof-theoretical consequence relations, denoted as $\models$ and $\vdash$, respectively, which are often interpreted as arguments (e.g., Purtill, 1979; Copi, 1982), have this structure.

The third type of arguments are the arguments that are constructed by *subordination*, such as the argument:

$Reason_1$.
So, $Reason_2$.
So, *Conclusion.*

This argument structure is most common. For instance, in Lin and Shoham's Argument Systems (Lin and Shoham, 1989; Lin, 1993) and Vreeswijk's Abstract Argumentation Systems (Vreeswijk, 1991, 1993),[4] arguments are explicitly constructed by subordination. Also the proofs of several proof theories for Propositional or First-Order Predicate Logic have this structure. Less obviously, this structure is also at the heart of Reiter's Default Logic (Reiter, 1980, 1987),[5] Bondarenko *et al*.'s Assumption-Based Framework for Non-Monotonic Reasoning (Bondarenko *et al*., 1993), and Loui and Chen's Argument Game (Loui and Chen, 1992). Pollock's linear arguments in his Theory of Defeasible Reasoning (1995, p. 39)[6] can be regarded as having this structure.[7]

---

[2] See also chapter 4, section 4.2.
[3] See, e.g., Van Dalen (1983) or Davis (1993).
[4] See also chapter 4, section 5.2.
[5] See also chapter 4, sections 3.1, 4.2 and 5.2.
[6] See also chapter 4, section 4.2.
[7] Pollock (1995, p. 39) defines linear arguments as finite sequences of sentences, each of which is either a premise or supported by a previous member of the sequence. The structure of linear arguments is not only ambiguous, as Pollock (1995, p. 87) notes, but is somewhat less expressive than that of subordinated arguments, because it cannot distinguish different occurrences of the same sentence in an argument. For instance, the arguments $\{\{\{\{A\}\} \rightarrow B\}\} \rightarrow C$ and $\{\{\{\{A\}\} \rightarrow B, A\}\} \rightarrow C$ in CumulA both correspond to the linear argument A, B, C.

The fourth type of arguments are the arguments that are constructed by both *subordination and coordination* of arguments, for instance:

Subreason$_{11}$, *Subreason*$_{12}$; *Subreason*$_{21}$, *Subreason*$_{22}$.
So, *Conclusion*.

This is the argument structure that is used in CumulA. In the argumentation theory of Van Eemeren and Grootendorst (Van Eemeren *et al.*, 1981, 1987), real-life arguments are reconstructed and evaluated using the mentioned argument structure.[8] Van Eemeren and Grootendorst have included both subordination and coordination in their model since both can be found in argumentative texts. In the next section, we argue for the need of coordination, especially for defeasible arguments because of defeat by pararallel strengthening and the accrual of reasons.

We mention a fifth type of argument structure that occurs, for instance, in natural deduction proofs of Propositional and First-Order Predicate Logic, and in Pollock's Theory of Defeasible Reasoning (Pollock, 1987-1995): *arguments with suppositions*. For instance, such arguments occur if the natural deduction rule of inference $\rightarrow$-Introduction is used in a proof or argument:

A proof of Q with suppositions in a set $\mathsf{S} \cup \{Q\}$ can be extended to a proof of $P \rightarrow Q$ with suppositions in the set S.

Here, a proof is considered relative to a set, the suppositions of the proof. The rule of inference$\rightarrow$-Introduction above shows that the set of suppositions can change. After the introduction of $P \rightarrow Q$, the supposition Q can be withdrawn.

If one reads 'argument' instead of 'proof', this rule of inference becomes a type of argument construction, as Pollock does. To include this type of argument construction in his argumentation model, Pollock (1995, p. 86ff.) constructs arguments not from sentences (as in CumulA), but from sentences relative to a set of suppositions, formally an ordered pair of a sentence and a set of sentences (P, S). For instance, the rule of inference $\rightarrow$-Introduction becomes: [9]

An argument supporting (Q, S $\cup$ {Q}) can be extended to an argument supporting ($P \rightarrow Q$, S).

We have not included this type of argument in CumulA for two reasons. First, we think that the intuition of an argument without suppositions is easier to grasp than the intuition of an argument with suppositions. Whereas arguments without suppositions can be thought of as consisting of steps that represent the support of a

[8] The terminology of Van Eemeren and Grootendorst differs from ours. Their multiple arguments correspond to CumulA's coordinated arguments (cf. chapter 5, note 7).

[9] We paraphrase Pollock's 'rule of inference graph formation' called conditionalization (Pollock, 1995, p. 90).

state of affairs (expressed by a sentence) by another state of affairs (expressed by another sentence), arguments with suppositions cannot be thought of that way. This is due to the fact that some

> '… natural deduction rules have an indirect, even quasi-metalogical character' (Haack, 1978, p. 19).

This does of course not diminish the importance of the arguments with suppositions based on natural deduction rules, and their role in argumentation certainly deserves further study.

Second, arguments with suppositions behave unexpectedly if they are defeasible, as Vreeswijk (1993, p. 185ff.) shows. He gives a technical example in which arguments that should be undefeated nevertheless become defeated if the rule of →-Introduction is adopted. Vreeswijk's conclusion is that it is best to leave arguments with suppositions out of theories of argumentation with defeasible arguments for now until we have a better understanding of the behavior of more simply structured defeasible arguments. Since, to the best of our knowledge, the problems pointed out by Vreeswijk have not been solved, we have adopted the same conclusion.

2 Argument structure and defeat

The structure of an argument can determine whether an argument is defeated. In this section, we treat different types of structure-based defeat, as they are found in existing argumentation models. We show how the types of defeat can be distinguished in CumulA.

The first and simplest type of structure-based defeat is the trivial type of *no defeat* at all. The prototypical examples of argumentation models that have no defeat are the classical deductive logics, such as Propositional and First-Order Predicate Logic. In CumulA, an argumentation theory has no defeat if it has no defeater schemes.

The second type of structure-based defeat is *sentence-type defeat*. The defeat of an argument is of sentence-type if the defeat depends on sentences occurring in the argument. For instance, an argument

> *Reason*.
> So, *Conclusion.*

might be defeated because of an (undefeated) statement that denies the conclusion, such as:

> *Not_conclusion.*

This is a case of sentence-type defeat: any argument containing the sentence *Conclusion* is defeated if the statement *Not_conclusion* is undefeated. A defeater scheme representing this in CumulA has the form

Not_conclusion [∗*Conclusion*].

The challenged argument scheme ∗*Conclusion* has any argument with conclusion *Conclusion* as an instance. If any argument with conclusion *Not_conclusion* challenges any argument with conclusion *Conclusion*, this would be represented by the defeater scheme

∗*Not_conclusion* [∗*Conclusion*].

We say that the two mentioned defeater schemes are of sentence-type, which means that all their argument schemes have a statement as an instance. An argumentation theory has sentence-type defeat if it has sentence-type defeater schemes.

Argumentation models with sentence-type defeat are Poole's Logical Framework for Default Reasoning (Poole, 1988), and Lin and Shoham's Argument Systems (Lin and Shoham, 1989; Lin, 1993). Also Dung's Argumentation Frameworks (Dung, 1993, 1995) can be regarded as having sentence-type defeat since all arguments are structureless.

Bondarenko *et al*.'s Assumption-Based Framework for Non-Monotonic Reasoning (Bondarenko *et al.*, 1993) describe a special kind of sentence-type defeat, that we call *assumption-type defeat*. There is a special set of assumptions, that can be used as premises of arguments. If there is an undefeated argument that has the denial of an assumption as its conclusion, all arguments with that assumption as a premise are defeated. A defeater scheme representing this in CumulA has the form

∗*Not_assumption* [*Assumption*].

This defeater scheme has no consequences for arguments that do not have *Assumption* as a premise, even if *Assumption* occurs in the argument elsewhere. A sentence-type defeater scheme, as the one above, that has only statements as challenged arguments, is of assumption-type. An argumentation theory has assumption-type defeat if it has defeater schemes of assumption-type.

The third type of structure-based defeat is *step-type defeat*. The defeat of an argument is of step-type if the defeat depends on a step occurring in the argument. For instance, an argument

Reason.
So, *Conclusion.*

might be defeated because there is an (undefeated) statement that does not deny the conclusion, but undercuts the argument step (cf. chapter 5, section 3.1):

Undercutter

This is a case of step-type defeat: any argument containing the argument step ‘*Reason*. So, *Conclusion*’ is defeated if the conclusion *Undercutter* is justified. A defeater scheme representing this in CumulA has the following form:

$*Undercutter$ [{{$*Reason$}} $\rightarrow$ *Conclusion*].

Another example of step-type defeat is rebuttal (cf. chapter 5, section 3.2): an argument

$Reason_1$.
So, *Conclusion.*

is defeated because there is an (undefeated) argument that supports the denial of its conclusion:

$Reason_2$.
So, *Not_conclusion.*

Any argument containing the step ‘$Reason_1$. So, *Conclusion*’ is defeated if an argument containing the step ‘$Reason_1$. So, *Conclusion*’ is undefeated. A defeater scheme representing this in CumulA has the following form:

{{$*Reason_2$}} $\rightarrow$ *Not_conclusion* [{{$*Reason_1$}} $\rightarrow$ *Conclusion*]

The latter two defeater schemes are of step-type: all their argument schemes have a single-step argument as an instance that is not of sentence-type. An argumentation theory has step-type defeat if it has step-type defeater schemes.

The fourth type of structure-based defeat is *composite-type defeat*. We speak of composite-type defeat if the defeat of an argument depends on a composite structure occurring in the argument. In chapter 5, sections 3.3 and 3.4, we discussed two kinds of composite-type defeat: defeat by sequential weakening and defeat by parallel strengthening. We recall that in defeat by sequential weakening an argument is defeated because it ends in some sequence of steps. A defeater scheme representing that any argument ending with the two-step sequence ‘*Reason*. So, $Conclusion_1$. So, $Conclusion_2$’ is always defeated has the following form:

[{{{{$*Reason$}} $\rightarrow$ $Conclusion_1$}} $\rightarrow$ $Conclusion_2$]

In defeat by parallel strengthening an argument is defeated because some argument that has narrowings (chapter 5, section 2.4) is undefeated. A defeater representing that any argument in which two reasons $Reason_1$ and $Reason_2$ support the conclusion *Conclusion* defeats any argument in which the reason $Reason_3$ supports *Not_conclusion* has the following form:

$$\{\{*Reason_1\}, \{*Reason_2\}\} \to Conclusion\ [\{\{*Reason_3\}\} \to Not\_conclusion]$$

The latter two defeaters are of composite-type, meaning that they are neither of sentence-type nor of step-type.[10] An argumentation theory has composite-type defeat if it has composite-type defeater schemes.

Most existing argumentation models do not have composite-type defeat. An exception is Vreeswijk's Abstract Argumentation Systems (Vreeswijk, 1991, 1993). In Vreeswijk's formalism defeat depends on a conclusive force relation on full arguments. However, since Vreeswijk only uses subordination to construct composite arguments and no coordination, his formalism only can model defeat by sequential weakening and not defeat by parallel strengthening.

Defeat by parallel strengthening requires the coordination of arguments. It is based on the natural idea of accrual of reasons:[11] A conclusion can be better supported if there are more independent reasons for it. Although several people have made the point that reasons can accrue,[12] it remains controversial.

For instance, Pollock (1991a, 1995, pp. 101-102) explicitly argues against accrual. He thinks accrual is a natural idea, but then gives an example that makes him doubt that reasons accrue. The example goes as follows. If someone testifies that the president of Slobovia has been assassinated, that is a reason that the president is assassinated. Accrual would imply that testimonies of different people make the fact that the president is assassinated more credible. Pollock points out that this does not generally hold and depends on contingent facts. For instance, if testimonies are indeed independent, they make the president's assassination more credible. However, the testimonies are not necessarily independent: we can imagine a community in which people only confirm each other's lies. In that case, more reasons based on testimonies do not give increasing support to the president's assassination: more than one testimony would even make the assassination unjustified.[13]

---

[10] Defeater schemes of composite-type should not be confused with compound defeater schemes. Compound defeater schemes are defeater schemes that contain more than one challenging or more than one challenged argument scheme (chapter 5, sections 3.5 and 3.7). See also the next section on individual and groupwise defeat.

[11] Pollock (1991, p. 51) uses this terminology.

[12] Chronologically: Naess (1978) in argumentation theory, Hage (1991) in legal reasoning, Pinkas (1991) in neural computing, Brewka and Gordon (1994) and Gabbay (1994, pp. 196-198) in formal logic, Visser (1995, p. 177) in AI and law.

[13] A similar, more realistic, example is the following, by Henry Prakken. John likes to walk if it is Sunday. John does not like to walk if it is either hot or raining. If it is either hot or

As a solution, Pollock proposes that different independent reasons for a conclusion are subsumed in a new composite reason. In our opinion, this approach probably can be made to work - Pollock does not give details. However, the example does not *necessitate* Pollock's approach, while the approach does throw away the intuitively attractive idea of accrual of reasons. Both in chapter 2 on Reason-Based Logic and in chapter 5 on CumulA, we have presented formalisms that capture accrual and still can deal with examples such as Pollock's. For instance, Pollock's example is captured in CumulA by the following compound defeater scheme:

$[\{\{*\text{Testimony}_1\}, \{*\text{Testimony}_2\}\} \rightarrow \text{Assassination}]$

Moreover, properties characteristic for accrual, such as the property that if a narrowing of an argument is undefeated, the argument itself is undefeated (chapter 5, section 4.1), and the property that, if the pros outweigh the cons, additional pros do not change the balance (chapter 2, section 5), can easily be overlooked.

3 Individual and groupwise defeat

The defeat of an argument often depends on other arguments. Mostly the defeat of an argument depends on one other argument, but not always. In this section, we distinguish argumentation models by the number of arguments that determine defeat.

First, the defeat of an argument can depend only on itself, and not on any other argument. We call this *self-defeat*. For instance, an argument that has a contradiction as its conclusion often is considered defeated, for instance in Lin and Shoham's Argument Systems (Lin and Shoham, 1989; Lin, 1993). In CumulA, this could be represented by a defeater scheme of the following form:

$[*\textit{Contradiction}]$

Another example is an argument that is defeated because it contains some sequence of steps, as in defeat by sequential weakening (chapter 5, section 3.3). If an argumentation theory has defeater schemes, the instances of which have no challenging and one challenged argument, we say the argumentation theory has self-defeat.

---

raining on Sunday, he does not like to walk. If it is hot and raining on Sunday, he likes to walk. The difficulty is here that the reasons 'It is hot' and 'It is raining' together are apparantly weaker, in contrast with the principle of accrual. Since we choose to keep the intuitively attractive principle of accrual, we propose to deal with this example by considering 'It is hot and raining' as a *new* reason, and not only as the coordination of two reasons.

Second, the defeat of an argument can depend on one other undefeated argument. We call this *simple defeat*. Examples are arguments that are defeated by an undercutter or by a rebutter, as distinguished in Pollock's Theory of Defeasible Reasoning (Pollock, 1987-1995). In CumulA, defeat by an undercutter or rebutter is represented by defeater schemes, such as the following two:

$*Undercutter$ [{{$*Reason$}} → $Conclusion$]
{{$*Reason_2$}} → $Not\_conclusion$ [{{$*Reason_1$}} → $Conclusion$]

Both defeater schemes are simple since their instances have at most one challenging and at most one challenged argument (chapter 5, section 3.7). If an argumentation theory has simple defeater schemes, we say it has simple defeat.

Third, the defeat of an argument can depend on more than one undefeated argument. We call this *left-compound defeat* (because of the form of the corresponding defeater schemes). An example is an argument that is defeated because its conclusion conflicts with the conclusion of other arguments, as for instance in Poole's Logical Framework for Default Reasoning (Poole, 1988) and Lin and Shoham's Argument Systems (Lin and Shoham, 1989; Lin, 1993). If $Conclusion_1$, … $Conclusion_{n-1}$ and $Conclusion_n$ are conflicting, this can in CumulA be represented by a defeater scheme of the following form:

$*Conclusion_1$, …, $*Conclusion_{n-1}$ [$*Conclusion_n$]

This defeater scheme is left-compound since its instances have more than one challenging argument (chapter 5, section 3.7). If an argumentation theory has left-compound defeater schemes, we say it has left-compound defeat.

Fourth, the defeat of an argument can depend on other defeated arguments. We call this *right-compound defeat*. An example is an argument that is defeated together with other arguments because their conclusions are conflicting, as the collective defeat of arguments in Pollock's Theory of Defeasible Reasoning (Pollock, 1987-1995). If $Conclusion_1$, … $Conclusion_{n-1}$ and $Conclusion_n$ are conflicting, this can in CumulA be represented by a defeater scheme of the following form:

[$*Conclusion_1$, …, $*Conclusion_{n-1}$, $*Conclusion_n$]

This defeater is right-compound since its instances have more than one challenged argument (chapter 5, section 3.7). If an argumentation theory has right-compound defeater schemes, we say it has right-compound defeat.

4 Triggers of defeat

Argumentation models can differ in the way the defeat of arguments is triggered. Two triggers of defeat can be distinguished: inconsistency and counterarguments. We call the resulting types of defeat inconsistency-triggered and counterargument-triggered defeat, respectively.[14]

Inconsistency-triggered defeat has the longest tradition and is related to the early work on nonmonotonic reasoning. Its basic intuition is that the defeat of arguments is at heart the maintenance of the consistency of argument conclusions. Many variants have been proposed. For instance, one of a (minimal) set of arguments with conflicting arguments can be considered defeated, as in Poole's Logical Framework for Default Reasoning (Poole, 1988) and Lin and Shoham's Argument Systems (Lin and Shoham, 1989; Lin, 1993). If $Conclusion_1$, … $Conclusion_{n-1}$ and $Conclusion_n$ are conflicting, this can in CumulA be represented by n (left-compound) defeater schemes of the following form:

$*Conclusion_1, \ldots, *Conclusion_{i-1}, *Conclusion_{i+1}, \ldots, *Conclusion_n$
$[*Conclusion_i]$

This leads to indeterministic defeat since each of these defeaters represents an arbitrary choice of a defeated argument (chapter 5, section 3.5).[15] In Vreeswijk's Abstract Argumentation Systems (Vreeswijk, 1991, 1993), the choice of a defeated argument is restricted by a conclusive force relation: an argument in a minimal set of arguments with conflicting conclusions cannot be considered defeated if it has stronger conclusive force than one of the other arguments in the set.

If an argumentation theory has defeater schemes the instances of which consist of arguments with conflicting conclusions (with respect to some appropriate sense of inconsistency), we say the argumentation theory has inconsistency-triggered defeat.

Counterargument-triggered defeat is based on another intuition: defeat is the result of arguments challenging other arguments. The purest version of counterargument-triggered defeat is Dung's formalism of Argumentation Frameworks (Dung, 1993, 1995). Dung studies a binary attack relation between arguments. In CumulA, his attacks can be represented as defeaters of the following form:

$Argument_1\ [Argument_2]$

---

[14] The distinction between inconsistency-triggered and counterargument-triggered defeat corresponds to Verheij's (1995a, b) distinction between indirect and direct defeat.

[15] As a result, indeterministic defeat leads to multiple extensions, as in many models of nonmonotonic reasoning. Cf. the overviews by Ginsberg (1987), Lukaszewicz (1990) and Gabbay *et al.* (1994b). See also chapter 5, section 6.2.

Attacks are represented as defeaters and not as defeater schemes since Dung treats arguments as structureless objects. As a result his arguments correspond to statements in CumulA. If an argumentation theory has defeater schemes the instances of which do not consist of arguments with conflicting conclusions (with respect to some appropriate sense of inconsistency), we say the argumentation theory has counterargument-triggered defeat. Clearly, general argumentation theories in CumulA have counterargument-based defeat.

In a way, counterargument-triggered defeat is more general than inconsistency-triggered defeat. Whereas inconsistency-triggered defeat can naturally be captured as a special case of counterargument-based defeat (as in the examples above), not all counterargument-triggered defeat can as naturally be captured as a special case of inconsistency-triggered defeat.

The distinction between inconsistency-triggered and counterargument-based defeat can be recognized if one considers rebutting and undercutting defeat. Rebutting defeat is by its nature an example of inconsistency-triggered defeat, but can as we have seen naturally be captured in the defeater schemes CumulA, which has counterargument-triggered defeat. Undercutting defeat is by its nature an example of counterargument-triggered defeat, and can naturally be captured in CumulA's defeater schemes, but not as naturally in inconsistency-triggered defeat.

For instance, Vreeswijk (1993, pp. 51-53) claims that it is possible to incorporate undercutting defeat in his Abstract Argumentation Systems, which have inconsistency-triggered defeat. However, in order to incorporate undercutting defeat, Vreeswijk has to adapt his argumentation model, as follows. He introduces a defeasible conditional > in his language. In a case of undercutting defeat, Vreeswijk forces an inconsistency between the conditional and its negation. The use of defeasible conditionals is a fine approach to undercutting defeat, and is very similar to the approach of Reason-Based Logic (chapter 2), but requires an adaptation of the formalism. Moreover, Vreeswijk hinges on two thoughts: he incorporates undercutting defeat using defeasible conditionals and rebutting defeat using argument defeat. However, we have seen that it is possible to capture both undercutting and rebutting defeat using defeasible conditionals (as for instance in Reason-Based Logic), and using argument defeat (as for instance in CumulA).

5 Directions of argumentation

Argumentation models can differ in the direction of argumentation they describe. We distinguish static, forward, backward and bidirectional argumentation.

Static argumentation occurs in argumentation models that do not treat argumentation as a process. No sequences of stages are considered, but only stages that are in some sense maximal. The extensions of Reiter's Default Logic (Reiter, 1980, 1987) and Poole's Logical Framework for Default Reasoning (Poole, 1988) can be regarded as such special stages.

Forward argumentation is the most common among existing argumentation models. Argumentation starts from a fixed set of premises. Arguments are constructed by adding forward steps. In forward argumentation, the goal is to find conclusions supported by arguments with given premises. For instance, Lin and Shoham's Argument Systems (Lin and Shoham, 1989; Lin, 1993), Vreeswijk's Abstract Argumentation Systems (Vreeswijk, 1991, 1993) and Bondarenko *et al*.'s Assumption-Based Framework for Non-Monotonic Reasoning (Bondarenko *et al.*, 1993) are models of forward argumentation. In CumulA, forward argumentation means that a line of argumentation only contains stages with premises in a fixed set.

Backward argumentation is less common. Argumentation starts from a set of conclusions. Arguments are constructed by adding backward steps. In backward argumentation, the goal is to find premises for arguments supporting given conclusions. For instance, Loui and Chen's Argument Game (Loui and Chen, 1992)[16] is a model with backward argumentation. In CumulA, backward argumentation means that a line of argumentation only contains stages with conclusions in a fixed set.

Bidirectional argumentation is the natural generalization of forward and backward argumentation. Argumentation does not start form a fixed set of premises or conclusions. Arguments are both forwardly and backwardly constructed. In bidirectional argumentation, the goal is neither only to find conclusions nor only to find premises, but a mixture of both. Except for CumulA, we know of no argumentation model of bidirectional argumentation.[17]

6 Capturing elements of argumentation models in CumulA

In the previous sections, we have discussed several ways to distinguish argumentation models. We explained how these distinctions can be made for CumulA argumentation theories. To be able to use the distinctions to compare existing argumentation models, we show how elements of a number of major argumentation models can be captured in argumentation theories of CumulA. We stress that we do not give formal relations between argumentation models and CumulA's argumentation theories. The presented argumentation theories capturing elements of existing argumentation models are meant to illustrate CumulA and our views on other argumentation models, and not to show strict formal relations.

Our selection of argumentation models is influenced by our focus, as made explicit by the CumulA model. Each selected argumentation model has been influential, or shows a specific characteristic of argumentation that falls within our focus. We have selected Propositional Logic, Poole's Logical Framework for

---

[16] Recently, a variant of Loui and Chen's Argument Game has been implemented by Kang.

[17] Pollock (1995, p. 153) describes forward and backward argumentation in another sense: he keeps both allowed premises and desired conclusions fixed. In bidirectional argumentation in our sense, neither premises nor conclusions are fixed.

Default Reasoning (Poole, 1988), Lin and Shoham's Argument Systems (Lin and Shoham, 1989; Lin, 1993), Reiter's Default Logic (Reiter, 1980, 1987), Pollock's Theory of Defeasible Reasoning (Pollock, 1987-1995), Vreeswijk's Abstract Argumentation Systems (Vreeswijk, 1991, 1993), Bondarenko *et al*.'s Assumption-Based Framework for Non-Monotonic Reasoning (Bondarenko *et al.*, 1993), Dung's Argumentation Frameworks (Dung, 1993, 1995), and Loui and Chen's Argument Game (Loui and Chen, 1992).[18]

We do not discuss all argumentation models in full detail, but capture elements that fall within our focus in CumulA. Some acquaintance with the discussed argumentation models is assumed.

6.1 Propositional Logic

We have selected Propositional Logic as an example of an argumentation model without defeat. An argumentation theory capturing elements of Propositional Logic in CumulA can be defined as follows:

Language = L_{PL}, the language of Propositional Logic.
Rules = $\{\{Sentence_1, \ldots, Sentence_n\} \rightarrow Sentence_{n+1} \mid$
$Sentence_1, \ldots, Sentence_n \models_{PL} Sentence_{n+1}\}$,
where $\models_{PL}$ denotes the consequence relation of Propositional Logic.
DefeaterSchemes = $\emptyset$.

The rules of the argumentation theory correspond to logical consequence in Propositional Logic. There are no defeater schemes.

Mostly only single-step arguments are considered, although proof theories for Propositional Logic can be interpreted as descriptions of subordinated arguments from a restricted set of rules. Accounts of Propositional Logic normally do not describe a counterpart of our lines of argumentation. Only maximal sets of conclusions from a set of premises are considered. These are similar to CumulA's forward extensions (restricted to single-step arguments).

This example shows that it is not necessary to explicitly distinguish classes of strict and defeasible arguments, as is done in many argumentation models, e.g. in Lin and Shoham's Argument Systems (Lin and Shoham, 1989; Lin, 1993) and Vreeswijk's Abstract Argumentation Systems (Vreeswijk, 1991, 1993). If required,

---

[18] Obvious omissions are the models of Nute (1988), Geffner and Pearl (1992), Simari and Loui (1992), Gordon (1993a, 1993b, 1995), Lodder and Herczog (1995), extending the work of Hage *et al*. (1994), and Prakken and Sartor (1996). All describe significant research, relevant for argumentation, but with a focus different from CumulA's. Nute focuses on a Prolog implementation, Geffner and Pearl on integration of argumentation and the so-called ε-semantics, Simari and Loui on the mathematics of argumentation and specificity, Gordon on dialogue in legal argumentation, Lodder and Herczog on dialogues and commitment, and Prakken and Sartor on defeasible priorities.

an argumentation theory can incorporate a set of arguments that cannot be defeated because the theory does not have defeater schemes that could cause their defeat.[19]

6.2 Poole's Logical Framework for Default Reasoning

We have selected Poole's Logical Framework for Default Reasoning since it is the purest example of consistency maintenance. An argumentation theory capturing elements of Poole's Framework in CumulA can be defined as follows:

Language = L_{PL}, the language of Propositional Logic.
Rules = $\{\{Sentence_1, \ldots, Sentence_n\} \rightarrow Sentence_{n+1} \mid Sentence_1, \ldots, Sentence_n \models_{PL} Sentence_{n+1}\}$,
where $\models_{PL}$ denotes the consequence relation of Propositional Logic.
DefeaterSchemes = $\{Sentence_1, \ldots, Sentence_{n-1}\ [Sentence_n] \mid Sentence_1, \ldots, Sentence_{n-1}, Sentence_n \models_{PL} \bot\}$,
where $\bot$ denotes contradiction in Propositional Logic.

The rules correspond to ordinary logical consequence in Propositional Logic, as in the argumentation theory for Propositional Logic above. The defeater schemes say that an argument is challenged by other arguments if the argument's conclusion is inconsistent with the conclusions of the other arguments.

In Poole's Framework, only single-step arguments are considered. Poole's Framework does not contain a counterpart of our lines of argumentation. Poole's extensions are similar to CumulA's forward extensions.

6.3 Lin and Shoham's Argument Systems

Lin and Shoham's Argument Systems are related to Poole's Logical Framework for Default Reasoning, since both deal mainly with consistency maintenance. We have selected Lin and Shoham's Argument Systems, since in this argumentation model it is recognized that the defeat of arguments can be studied independent of the specific language and argument rules, and that for the study of argument defeat it is useful to consider special sets of structured arguments, such as sets of arguments closed under initials.

An argumentation theory capturing elements of Lin and Shoham's Argument Systems in CumulA can be defined as follows:

Language = $Atoms \cup \neg Atoms$,
where *Atoms* is any set and $\neg Atoms$ is the set $\{\neg Atom \mid Atom$ is an element of $Atoms\}$ (disjoint from *Atoms*).
Rules is any set of rules of the language.

[19] If moreover strict arguments always should defeat defeasible arguments in case of a conflict, additional defeaters are required.

DefeaterSchemes = {∗*Atom* [∗¬*Atom*], ∗¬*Atom* [∗*Atom*] | *Atom* is an element of *Atoms*}.

Lin and Shoham abstract from the language used. It is a set of sentences closed under negation. The set of rules is arbitrary. The defeater schemes represent that an argument challenges another if it has opposite conclusion.

Lin and Shoham consider subordinated arguments, and forward lines of argumentation.

6.4 Reiter's Default Logic

Reiter's Default Logic is selected since it rightly remains influential. It should be regarded as an argumentation model avant la lettre. An argumentation theory capturing elements of Reiter's Default Logic in CumulA can be defined as follows:

Language = L_{PL}, the language of Propositional Logic.
Rules ⊇ {{$Sentence_1$, …, $Sentence_n$} → $Sentence_{n+1}$ | $Sentence_1$, …, $Sentence_n \models_{PL} Sentence_{n+1}$},
where $\models_{PL}$ denotes the consequence relation of Propositional Logic.
DefeaterSchemes ⊆
{∗¬*Justification* [{{∗$Condition_1$, …, ∗$Condition_n$}} → *Conclusion*] | {$Condition_1$, …, $Condition_n$} → *Conclusion* is an element of *Rules*}.[20]

For convenience, we restricted the language to Propositional Logic. The set of rules is a superset of the set of rules corresponding to ordinary logical consequence. As in Default Logic, rules have so-called justifications. A rule can only be used if its justification is not denied. This leads to defeater schemes of a special form: an argument justifying the negation of a justification of some rule challenges an argument that ends with a step corresponding to the rule. So, a default $Condition_1$, …, $Condition_n$: $Justification_1$, …, $Justification_m$ / *Conclusion* of Default Logic corresponds to a rule {$Condition_1$, …, $Condition_n$} → *Conclusion* in *Rules* and defeater schemes ∗¬$Justification_i$ [{{∗$Condition_1$, …, ∗$Condition_n$}} → *Conclusion*], for i = 1 to m, in *DefeaterSchemes*. (So, defaults that only differ in their justifications are not distinguished.)

Reiter's Default Logic implicitly describes subordinated arguments and no forward lines of argumentation. Reiter's extensions are similar to CumulA's forward extensions.

[20] Lin and Shoham (Lin and Shoham, 1989; Lin, 1993) and Dung (1995) show how Reiter's (1980, 1987) Default Logic can be translated to their argumentation models. In contrast with us, they also prove formal relations.

6.5 Pollock's Theory of Defeasible Reasoning

Pollock's Theory of Defeasible Reasoning is probably the most worked-out argumentation model. It has been developed and adapted since 1987. An argumentation theory capturing elements of Pollock's theory in CumulA can be defined as follows:

Language = L_{PL}, the language of Propositional Logic.
Rules $\supseteq$ {{*Sentence*$_1$, …, *Sentence*$_n$} $\rightarrow$ *Sentence*$_{n+1}$ |
Sentence$_1$, …, *Sentence*$_n$ $\models_{PL}$ *Sentence*$_{n+1}$},
where $\models_{PL}$ denotes the consequence relation of Propositional Logic.
CollectiveDefeat $\subseteq$ *DefeaterSchemes* $\subseteq$ *Undercutters* $\cup$ *Rebutters* $\cup$ *CollectiveDefeat*,
where
CollectiveDefeat =
{[{$*$*Subreason*$_{11}$, …, $*$*Subreason*$_{n1}$} $\rightarrow$ *Conclusion*$_1$, …,
{$*$*Subreason*$_{1m}$, …, $*$*Subreason*$_{nm}$} $\rightarrow$ *Conclusion*$_m$] |
Conclusion$_1$, …, *Conclusion*$_n$ is minimally inconsistent},
Undercutters =
{$*$*Conclusion*$_1$ [{$*$*Subreason*$_1$, …, $*$*Subreason*$_n$} $\rightarrow$ *Conclusion*$_2$]},
and
Rebutters =
{{$*$*Subreason*$_{11}$, …, $*$*Subreason*$_{n1}$} $\rightarrow$ *Conclusion*$_1$
[{$*$*Subreason*$_{12}$, …, $*$*Subreason*$_{n2}$} $\rightarrow$ *Conclusion*$_2$] |
Conclusion$_1$, *Conclusion*$_2$ $\models_{PL}$ $\perp$}.

Again, the set of rules is a superset of the rules corresponding to ordinary logical consequence. The defeater schemes are of three forms: those representing collective defeat (restricted to arguments with inconsistent conclusions), undercutting defeat, and rebutting defeat (see chapter 5, section 3.5, 3.1, and 3.2, respectively). Since Pollock uses collective defeat as a general means to preserve consistency, the set of defeater schemes is a superset of the set of defeater schemes representing collective defeat.

Pollock describes subordinated arguments and forward lines of argumentation.

6.6 Vreeswijk's Abstract Argumentation Systems

Vreeswijk's Abstract Argumentation Systems have been selected since Vreeswijk's argumentation model has influenced the development of CumulA (see chapter 5). Vreeswijk's model can be regarded as a refinement of Lin and Shoham's Argument Systems. An argumentation theory capturing elements of Vreeswijk's Abstract Argumentation Systems in CumulA can be defined as follows:

Language is any set, containing $\perp$, denoting contradiction.

Rules is any set of rules in the language.
DefeaterSchemes ⊆ {$Argument_1$, …, $Argument_{n-1}$ [$Argument_n$] |
There is a rule {Conclusion($Argument_1$), …, Conclusion($Argument_n$)} → ⊥}.

Just as Lin and Shoham's Argument Systems and CumulA, Vreeswijk's model is independent of a specific language; Vreeswijk's language only contains a special element denoting contradiction. The set of rules is arbitrary. The defeater schemes of Vreeswijk's model represent that an argument is challenged by other arguments, if the argument's conclusion is inconsistent with the conclusions of the other arguments. The defeater schemes resemble those of the theory capturing elements of Lin and Shoham's Argument Systems. However, there are three differences. First, Vreeswijk notion of inconsistency is somewhat more general than Lin and Shoham's since it includes inconsistency of more than two arguments. Second, only a subset of the defeater schemes is used. Which defeater schemes are selected depends on Vreeswijk's conclusive force relation, included in each Abstract Argumentation System, in the following way: for arguments $Argument_1$, …, $Argument_n$, such that there is a rule {Conclusion($Argument_1$), …, Conclusion($Argument_n$)} → ⊥, the fact that for some i, 1 ≤ i ≤ n, $Argument_i$ has less conclusive force than $Argument_n$ implies that $Argument_1$, …, $Argument_{n-1}$ [$Argument_n$] is not in *DefeaterSchemes*.[21] Third, the defeater schemes corresponding to Vreeswijk's model are of composite-type, whereas those of Lin and Shoham's model are of sentence-type. This is the result of the fact that Vreeswijk's conclusive force relation is a relation between full arguments.

Vreeswijk's model describes subordinated arguments and forward lines of argumentation.

6.7 Bondarenko *et al.*'s Assumption-Based Framework

Bondarenko *et al.*'s Assumption-Based Framework for Non-Monotonic Reasoning have been selected since the formalism has a specific type of defeat, that is worth distinguishing: assumption-type defeat. An argumentation theory capturing this specific element of Bondarenko *et al.*'s Assumption-Based Framework in CumulA can be defined as follows:

Language = *Atoms* ∪ ¬*Atoms*,
where *Atoms* is any set and ¬*Atoms* is the set {¬*Atom* | *Atom* is an element of *Atoms*} (disjoint from *Atoms*).
Rules is any set of rules in the language.
DefeaterSchemes ⊆ {∗*Atom* [¬*Atom*], ∗¬*Atom* [*Atom*] | *Atom* is an element of *Atoms*}.

[21] It could be interesting to establish formal connections between properties of a conclusive force relation and those of the corresponding set of defeater schemes.

This theory is related to the one capturing elements of Lin and Shoham's Argument Systems. The language is a set closed under negation and the set of rules is arbitrary. However, the defeater schemes differ subtly from those in the theory capturing elements of Lin and Shoham's, in two ways. First, the challenged arguments in the instances of the defeater schemes are statements. As a result, argument defeat of a non-statement argument is always indirect (see chapter 5, sections 4.3 and 4.4), because of the defeat of a premise of the argument. The defeater schemes are of assumption-type (see section 2). Second, not all defeater schemes of the given form need to be included in the argumentation theory. If ∗*Atom* [¬*Atom*] (or ∗¬*Atom* [*Atom*]) is included, ¬*Atom* (or *Atom*, respectively) is called an assumption of the theory. Intuitively, an assumption can be the premise of an undefeated argument, unless its negation is justified.

Bondarenko *et al*.'s model implicitly describes subordinated arguments and forward lines of argumentation.

6.8 Dung's Argumentation Frameworks

Dung's Argumentation Frameworks have been selected since Dung has brought the abstract study of argumentation and defeat to its extreme. Dung notices that the basis of defeat is the attack relation between arguments. As a result, he focuses on that relation, independent of the structure of the arguments involved. This is an important step towards a better understanding of argumentation and defeat.

An argumentation theory capturing elements of Dung's Argumentation Frameworks in CumulA can be defined as follows:

Language is any set.
Rules = ∅.
DefeaterSchemes ⊆ {$Statement_1$ [$Statement_2$]}

As Lin and Shoham's Argument Systems, Vreeswijk's Abstract Argumentation Systems and CumulA, Dung's model is independent of a specific language. Moreover, Dung abstracts from the structure of arguments. As a result, the set of rules is empty. The defeater schemes - actually defeaters - are all simple defeaters.

Dung considers unstructured arguments, corresponding to CumulA's statements, and no lines of argumentation. Verheij (1996a) investigates the formal relations between Dung's model and the stages approach of CumulA.

6.9 Loui and Chen's Argument Game

Loui and Chen's Argument Game has been selected since it shows a characteristic of argumentation not found in any of the other discussed argumentation models: backward argumentation. The Argument Game is a two-player card game, designed as a model of argumentation. One of the players tries to justify a conclusion by means of an undefeated argument, the other tries to challenge the argument. As a result, the conclusion is fixed, while the premises vary throughout the game.

An argumentation theory capturing elements of Loui and Chen's Argument Game in CumulA can be defined as follows:

Language = *Atoms* ∪ ¬*Atoms*,
where *Atoms* is any set and ¬*Atoms* is the set {¬*Atom* | *Atom* is an element of *Atoms*} (disjoint from *Atoms*).
Rules is any set of rules in the language.
DefeaterSchemes ⊆ {∗*Atom* [∗¬*Atom*], ∗¬*Atom* [∗*Atom*] | *Atom* is an element of *Atoms*}

Surprisingly, this argumentation theory is the same as the one capturing elements of Lin and Shoham's Argument Systems. This shows that the underlying notions of argument and defeat are the same in both models. However, argumentation is different in both models, since Loui and Chen consider backward lines of argumentation. Moreover, other differences between the models have disappeared, since we only focus on the underlying model of argumentation, and have therefore abstracted from the game elements of the Argument Game, such as bidding and the different roles of the players.

The arguments of Loui and Chen's Argument Game are constructed by subordination. The game models backward lines of argumentation with a single fixed conclusion.

7 A comparison of argumentation models

After capturing elements of several argumentation models as argumentation theories in CumulA in the previous section, we now apply the distinctions discussed in the sections 1 to 5 to those argumentation theories. An overview is given in table 1. The table shows differences and similarities.

We have shown the generality of CumulA by capturing elements of selected argumentation models in CumulA. Previously, Lin and Shoham (Lin and Shoham, 1989; Lin, 1993) and Dung (1995) have captured other selections of argumentation models in their formalisms. We stress that, in contrast with us, they have also proven formal relations.

Lin (1993) has also classified formalisms of nonmonotonic reasoning, using a distinction based on intuition. He distinguished two classes, namely sentence-based and argument-based formalisms. His distinction seems to be close to our distinction of sentence-type and composite-type defeat. Interestingly, in a footnote, Lin (1993, note 1, p. 254) remarks that Default Logic (Reiter, 1980) should probably be classified in both categories. We are able to clarify the position of Default Logic by classifying it in the intermediate class of step-type defeat.

Argumentation model	Type of arguments	Argument structure and defeat	Individual or group-wise defeat	Trigger of defeat	Direction of argumentation
Propositional Logic	single-step or subordination	no defeat	no defeat	no defeat	static
Poole's Logical Framework for Default Reasoning	single-step	sentence-type defeat	self-defeat and left-compound defeat	inconsistency-triggered defeat	static
Lin and Shoham's Argument Systems	subordination	sentence-type defeat	simple defeat	inconsistency-triggered defeat	forward argumentation
Reiter's Default Logic	subordination	step-type defeat	simple defeat	counterargument-triggered defeat	static
Pollock's Theory of Defeasible Reasoning	subordination	sentence-type and step-type defeat	self-defeat, simple and right-compound defeat	counterargument-triggered defeat	forward argumentation
Vreeswijk's Abstract Argumentation Systems	subordination	composite-type defeat	self-defeat and left-compound defeat	inconsistency-triggered defeat	forward argumentation
Bondarenko *et al.*'s Assumption-Based Framework	subordination	assumption-type defeat	simple defeat	inconsistency-triggered defeat	forward argumentation
Dung's Argumentation Frameworks	statements	sentence-type defeat	simple defeat	counterargument-triggered defeat	static
Loui and Chen's Argument Game	subordination	sentence-type defeat	simple defeat	inconsistency-triggered defeat	backward argumentation
CumulA	subordination and coordination	all types	all types	counterargument-triggered defeat	bidirectional argumentation

Table 1: A comparison of argumentation models

Since we focused on the argumentation theories capturing elements of argumentation models in CumulA, we were able to establish a number of distinctions on formal grounds in contrast with Lin's distinction based on intuition. As a result, we have shown similarities and differences between the argumentation models.