DiaLaw

On legal justification and dialog games
DiaLaw

On legal justification and dialog games

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit Maastricht,
op gezag van de Rector Magnificus, Prof. dr. A.C. Nieuwenhuijzen Kraseman,
volgens het besluit van het College van Decanen,
in het openbaar te verdedigen
op vrijdag 5 juni 1998 om 14.00 uur

door

Adrianus Roelant Lodder
Promotor:
  Prof. dr. H.F.M. Crombag

Co-promotor:
  Dr. J.C. Hage

Beoordelingscommissie:
  Prof. mr. N.H.M. Roos (voorzitter)
  Dr. T.F. Gordon (German National Research Center for Information Technology)
  Prof. dr. H.J. van den Herik
  Prof. dr. A. Peczenik (Universitet i Lund)

ISBN 90-901721-0
Preface

This thesis is not about Intelligent Tutoring Systems in the domain of International Tax Law, as was originally intended. The present thesis reflects the research on the dialog game for legal justification DiaLaw, and is structured as follows. After introductory observations, the reasons why legal justification can best be modeled as a process, viz. dialogically are discussed. Subsequently the model DiaLaw is formally defined, and legal examples demonstrating the system are given. Next other models of dialogical argumentation are evaluated. Finally, it is argued that there is more to formal arguments than logical validity; rhetorical, psychological aspects of argument have to be taken into account, even in formal models.

Writing a thesis is a solitary activity. Yet the result cannot be achieved without the support of many people. I am glad to acknowledge some of these people here. Note that if you are not mentioned it cannot be derived a contrario that I do not owe you a thank-you as well.

My advisor Hans Crombag supported me, especially on the harder moments, and gave useful comments on the final version of my dissertation. My co-advisor Jaap Hage has been the actual supervisor, thanks to him I learned many things. His enthusiasm and dedication are admirable. Remarkably his commentary always followed quickly after I handed my writings over to him, often even the same day. Over the years we discussed frequently, not only justifying our positions in flaming dialogs, always convinced of our own position, but also exchanging trivia about our shared passion popular music.

Over five years I shared a room with Bart Verheij. We became an indistinguishable duo. After many wild plans, this year we finally co-authored two papers.

The discussions with my fellow dialog researcher Ronald Leenes were always inspiring. A pleasant working environment has been provided by the members of the department of Metajuridica, e.g., my dear squash mate Sef Janssen, Georges Span, and the secretariat. Irreplaceable has been Math Lieben. Really interesting to participate in was the SKBS.B3-project of which the members were (excluding those already mentioned above): Arie Hasman, Jan Dietz, Jaap van den Herik, Maarten van der Meulen, Ruud van der Pol, Bob Quast & the one who offered computer tips and was good at spoofing, Floris Wiesman.

My friends and family are precious to me. They gave me the necessary love, enjoyment, and distraction.

Finally, my beloved Aimée, after all night long discussions DiaLaw was given shape, numerous times you patiently listened when I practiced my presentations, and together with Manderijn and Mikesh you made our house into a very very very fine house ...
Aan Aimée
Contents

Chapter 1  Introduction

1. Artificial Intelligence and Law ......................................................... 2
2. Research questions .......................................................................... 3
3. Outline of the thesis .......................................................................... 5

Chapter 2  From law to DiaLaw
Why legal justification should be modeled dialogically

1. The product and the process of justification ...................................... 8
2. Justification as a product .................................................................. 10
3. The defeasible nature of legal justification ....................................... 11
   3.1 Exceptions to rules ....................................................................... 12
   3.2 Conflicting norms ......................................................................... 13
4. The open nature of law ..................................................................... 14
   4.1 Cases without an applicable norm ............................................... 14
   4.2 Vague legal language ................................................................... 15
5. The Münchhausen Trilemma .............................................................. 16
6. Justification as a process: a dialogical model .................................... 18
7. Dealing with defeasibility, open nature, and the Münchhausen Trilemma in a dialogical model ......................................................... 20
8. Justification of dialog rules ............................................................... 21
9. How pure is the procedure of legal justification? ............................. 22
10. Conclusion ....................................................................................... 25

Chapter 3  DiaLaw
A dialogical model for legal justification

1. Justification in DiaLaw ................................................................. 27
2. Basic concepts of DiaLaw ............................................................... 28
   2.1 The participants ........................................................................ 29
   2.2 The moves of the game .............................................................. 29
   2.3 Burden of proof ....................................................................... 31
   2.4 Commitment .......................................................................... 31
   2.5 The dialog rules .................................................................... 32
   2.6 Levels in the dialog ................................................................. 32
3. DiaLaw’s dialogical framework ......................................................... 33
   3.1 The language ........................................................................ 33
   3.2 The dialog move ..................................................................... 34
   3.3 The commitment store ........................................................... 35
Chapter 4  DiaLaw in action

1. The Tyrell case .............................................................. 63
2. The Chabot case ............................................................. 68
   2.1 On reasons to apply a rule .......................................... 69
   2.2 On applies, reason, and excluded .............................. 70
   2.3 On reasons and outweighs .......................................... 71
3. Concluding remarks ....................................................... 73

Chapter 5  Dialogical models of argumentation

1. The purpose of models .................................................... 75
2. A short sample dialog .................................................... 76
3. MacKenzie’s DC and Hamblin’s H ..................................... 79
   3.1 The moves ............................................................. 79
   3.2 Commitment ........................................................... 81
   3.3 Concluding remarks ................................................ 82
4. Rescher’s Dialectics and the formal elaboration by Brewka ........ 83
   4.1 The moves ............................................................. 83
   4.2 Dialog rules .......................................................... 84
   4.3 Commitment .......................................................... 86
   4.4 Concluding remarks ................................................ 88
5. Gordon’s Pleadings Game ................................................ 89
   5.1 An informal description of the Pleadings Game ............... 90
Chapter 6  What is an argument?
The logical and psychological perspective

1. Argumentation: two perspectives .............................................. 117
   1.1 The logical perspective .................................................. 117
   1.2 The psychological perspective ....................................... 118
2. Argumentation: two types .................................................. 119
   2.1 Rational and dia-rational argumentation ......................... 119
   2.2 (Di)ra-rational argumentation in case law ..................... 120
3. Towards combining the approaches ...................................... 122
5. Conclusion ........................................................................ 124

Chapter 7  In conclusion

1. The answers ...................................................................... 125
2. The contribution of the thesis ........................................... 126
3. Closing remarks .................................................................. 127

Appendix- Prolog code of DiaLaw ............................................. 129
Samenvatting (in Dutch) .......................................................... 137
References ........................................................................... 143
Author index ......................................................................... 151
Subject index ......................................................................... 153
Curriculum vitae .................................................................... 157
Contact information ................................................................ 159
Chapter 1

Introduction

...justice is a game¹

First there was the law, then came computers.

The connection between law and computers is studied in two different areas of research. In the first area legal aspects related to computers are studied, or—in modern terminology—the object of study is the legal side of information technology. Interesting questions are amongst others: whether electronic signatures have the same status as written ones; whether an electronic copy of a file that is displayed on screen can be an infringement of property right; how the privacy of citizens concerning personal data stored in government data bases can be guaranteed, and whether the government may interconnect data bases to obtain interesting new information; whether an internet-provider is responsible for the information their clients present on Web-pages. In a simple way, the connection between law and computers in the first area of research can be depicted as follows.

In the second area of research it is studied what opportunities the computer offers to the legal discipline. Do computers have something to offer to the law? Although it is doubtful that the computer will ever be able to fully replace human judges,² even the most old-fashioned jurist sees benefits of computers.³ The connection between law and computers in the second area of research that is known as Legal Informatics,⁴ can be depicted as follows.

---

¹ Lyrics from the 1975 album *Desire* by Bob Dylan.

² There is at least one person that has no serious doubts [Van den Herik, 1991].

³ For instance, there are data bases with large amounts of verdicts by the Dutch Supreme Court and lower courts, special standard files for warrants to appear or verdicts, and nowadays it is even possible to read a decision of, for example, the Supreme Court of the USA through internet a few minutes after the Court decided.

⁴ Similar fields are amongst others Medical Informatics for Medicine, and Computer Algebra for Mathematics. The term Legal Informatics is a translation of the Dutch term ‘Rechtsinformatie’. Mostly the term ‘Computers and Law’ is used, but it actually includes both connections between Computers and Law. At least at Liverpool they use the term Legal Informatics, viz. LIAL.
Most research that is conducted in the field of Legal Informatics belongs to a sub-field that is known as Artificial Intelligence and Law (AI & Law). This thesis belongs to the sub-field AI & Law.

1. **Artificial Intelligence and Law**

AI & Law is a rather new field that originated at the beginning of the 1970s. The developments were modest in the first years. The field really began to flourish in the 1980s. In 1981, a first international conference in this field was held in Firenze, Italy. The First International Conference on Artificial Intelligence and Law, organized by the International Association for AI & Law (IAAIL) was held in Boston, USA [ICAIL, 1987]. From the mid 1980s onwards the first books on AI & Law were published [e.g., Susskind, 1987; Gardner, 1987], and in 1986 research conducted at Imperial College, London, was reported on in the classic [Sergot et al., 1986]. Since 1988 annual international conferences were organized in the Netherlands by the foundation of legal knowledge based systems, JURIX, in 1997 for the tenth time. In 1992 an international journal, *Artificial Intelligence and Law* saw the light of day.

What has the research on AI & Law brought us so far?

---

5 Buchanan & Hoadrick [1970] wrote a paper that is considered the first in this field. In 1995, they were invited to The Fifth International Conference on AI & Law to celebrate the 25th anniversary of the appearance of their paper. They gave a presentation *Some speculations about Artificial Intelligence and Legal Reasoning (25 years later)*.

6 It was the First International Conference on Logic, Informatics, Law and had as special themes Deontic Logic, Computational Linguistics and Legal Information Systems. Consecutive conferences were held in Firenze in 1985, 1989, and 1993.

7 Consecutive conferences were held in Vancouver, Canada [ICAIL, 1989]; Oxford, UK [ICAIL, 1991]; Amsterdam, the Netherlands [ICAIL, 1993]; Maryland, USA [ICAIL, 1995]; and Melbourne, Australia [ICAIL, 1997].

8 Some other related conferences are: the First Annual Conference on Law and Technology, held in 1984 in Houston, USA, with as theme artificial intelligence, information retrieval, and legal reasoning [Walter, 1985]; The BILETA conferences, focusing on technology in legal education, held annually from 1986 in the UK; from 1988, also in the UK, the biannual National Conference on Law, Computers and AI, has been organized, in 1996 as the Fifth National/First European Conference on Law, Computers and AI; the biannual French-American Conference on AI & Law was launched in 1996.

9 An other international journal is *Information and Communications Technology Law*, formerly published under the name of *Law, Computers and Artificial Intelligence*. 
Introduction

- An impressive series of doctoral dissertations on various subjects [e.g., Gardner, 1984; Ashley, 1987; Sartor, 1989; Brantling, 1991; Walker, 1992; Span, 1992; Prakken, 1993; Wahlgren, 1993; Gordon, 1993b; Smith, 1994; Voermans, 1995; Valente, 1995; Van Kralingen, 1995; Visser, 1995; Den Haan, 1996; Verheij, 1996; Aleven, 1997; De Vey Mestdagh, 1997].


- An immense number of articles that appeared in journals and conference proceedings.

The above is meant as an impression of the AI & Law research, and is not intended to be an exhaustive account. An interesting fact is the large amount of recent Dutch dissertations in the AI & Law field.10

Researchers in the field of AI & Law try to describe the law and legal reasoning in such a way that the resulting models can be implemented in a computer program. To build good working applications is a first goal for AI & Law research. This line of research could be called practical or applied AI & Law.

Since computers are rigid, the models have to be precise. A question that arises is how legal reasoning can be described precisely if there is no consensus about what legal reasoning precisely is? Trying to answer this question is the second goal for AI & Law research. There is no pretension that one right answer can be given, but precise models (and computer programs) can contribute to a better insight into law and legal reasoning.11 This line of research could be called theoretical AI & Law or computational legal theory.12

Recently it was argued that both groups should cooperate more closely, in order to obtain theoretically-founded, good working systems [Oskamp, Tragter & Groendijk, 1995]. Although I emphasize the need to test theoretical research in practice [Verheij, Hage & Lodder, 1997], the current thesis belongs only to the second line of research.

2. Research Questions

The title of this thesis is DiaLaw, on legal justification and dialog game. DiaLaw is a dialogical model of legal justification.

Legal justification is an interesting object of study, because there is no generally accepted theory yet that satisfactorily explains this phenomenon. What makes a legal statement a justified one is a question that is answered in different, even contradictory,

10 This is not really surprising, because of the prominent role the Netherlands plays in the field of AI & Law. For instance, a third of the papers of the ICAIL-conference of 1995 had Dutch authors.

11 A similar answer can be given to the question how artificial intelligence can be studied, if we do not know yet what intelligence is? AI models contribute to a better insight into intelligence.

12 The insight is necessary for developing ‘intelligent’ legal knowledge-based systems that are also used in legal practice.
ways. An interesting opinion, for example, is that of the Dutch legal theorist Scholten [1974, p. 133] who claimed that a decision by a judge is acceptable if a judge can testify that s/he could not have decided differently. However, Scholten admits that it does not imply that this decision is the only possible one, for another judge could decide differently and testify the same. This shows that even a single theory can have contradictory elements.

Unger [1983] was very skeptical about available legal theories and claimed that they are inappropriate abstractions of the actual law: the theories do not represent the law, but the personal opinions of the author. Although maybe he exaggerates a little, he surely has a point. I believe that the law, i.e., legal justification, is too complex to be fully understood ever. However, even if theories of legal justification reflect the personal opinions of the author, they can nevertheless be appropriate representations. Theories on legal justification can help in obtaining a better understanding of the factors that play a role in legal justification, and insight into legal justification is important for any jurist.

The present research is an attempt to characterize legal justification. This characterization must lead to a model that is sufficiently precise to implement in a computer. In legal justification arguments play an important role; actually, arguments are crucial [cf. Bench-Capon, 1995]. The importance of arguments is expressed in the observation that was an inspiring starting point for this research [cf. Lodder, 1997c],

By convincing arguments, an ostensibly lost case can be won.

The purpose of the research is finding answers to the following questions:

- How can be determined whether a legal statement is justified, and are there criteria a legal statement must satisfy in order to be justified?
- What model of legal justification is satisfactory, and is it possible to define and implement this model?
- How do comparable models represent argumentation?
- What is acceptable as an argument in legal justification?

Since none of the currently available models were fully satisfactory to model legal justification, a new model had to be developed. The model DiaLaw is meant to serve this purpose and is central in this thesis.

DiaLaw is not an empirical model. Primarily, DiaLaw is an analytical model. It helps to obtain a better insight into legal justification in general, and the process of legal justification in particular. Secondary, DiaLaw is a normative model, for it offers guidelines of how legal justification should be conducted.

---

13 Hart [1961] made a similar observation, not concerning legal justification but the broader concept of law: "Few questions concerning human society have been asked with such persistence and answered by serious thinkers in so many diverse, strange and even paradoxical ways as the question ‘What is law?’"

14 For reasons of brevity and uniformity in the remainder of this work 'he' is used.
3. **Outline of the Thesis**

The thesis is structured as follows.

In the second chapter a theoretical foundation on which a model of legal justification can be built is provided. Both the product and the process of justification are evaluated. A dialogical model of legal justification is proposed.

In the third chapter the dialogical model of legal justification DiaLaw is defined completely. In addition to defining a dialogical framework that consists of the language, the dialog move, the dialog, the commitment store, and the dialog tree, rules are formulated that regulate the argumentation process. The implementation of DiaLaw is discussed in its main lines.

In the fourth chapter two legal cases are modeled in order to illustrate the working of DiaLaw. The first case is about illegally obtained evidence, and is used to show in one dialog several aspects of DiaLaw. The second case is about euthanasia on a merely mentally suffering patient. This case is used to show in several short dialogs the working of the more complicated rules of DiaLaw.

In the fifth chapter other dialogical models are discussed. Central in this chapter are the moves the players can perform, and the commitment the players obtain. Three models are extensively addressed, and compared to each other and DiaLaw: the dialectical model of the philosopher Rescher [1977], the dialog game DC of Mackenzie [1979a], and Gordon's Pleadings Game [1995].

In the sixth chapter two perspectives on argumentation, the logical perspective and the psychological perspective, are discussed and related to rational and a-rational (called dia-rational in the chapter's context) argumentation. It is argued that a model of justification should allow both rational and (di)a-rational argumentation. DiaLaw is such a model.

In the final chapter the research questions will be revisited and answered, and the contributions of the thesis are addressed. Moreover, directions for future research are indicated.
Chapter 2

From law to DiaLaw

Why legal justification should be modeled dialogically

And you know you're never sure
But you're sure you could be right

In 1992 Hage, Span & Loder opened their contribution to the JURIX conference with the following statement:

Although most legal reasoning takes place in the context of legal disputes in which two parties argue their positions in a dialog with the other party, formal models of legal reasoning usually resemble monological proofs.

Things have radically changed since then. Although at that time our characterization of formal models of legal reasoning was accurate, at present it no longer is. The number of papers that reported on formal models of legal reasoning based on a dialogical approach is impressive [e.g., Gordon, 1993a; Loui et al., 1993; Hage, Leenes & Loder, 1994; Loder & Herczog, 1995; Farley & Freeman, 1995; Prakken & Sartor, 1996; Kowalski & Toni, 1996; Nitta & Shibasaki, 1997]. Most papers describe dialogical models; in the next chapter a description of the model DiaLaw is given. The present chapter discusses the reasons why legal justification can best be modeled dialogically (see also Loder [1996b, 1997b]).

Legal justification is a special type of legal reasoning. The acceptability of a legal statement depends on the quality of its justification [Feteris, 1994, p. 1]. In this chapter it is argued that the quality of a justification can best be determined in a dialogical model of legal justification. The purpose of this chapter is to:

1. convince those that not yet have seen the benefits of dialogical models;
2. describe my motives for those that are already attached to dialogical models;
3. justify the development of DiaLaw.

Most of the reasons for the claim that legal justification needs to be modeled dialogically are not new. The originality of the present chapter lies particularly in the context within which the reasons are discussed.

The chapter is structured as follows: section 1 provides an introduction to the difference between the product and process of justification. Subsequently an account is given of justification as a product, in a deductive model (section 2). Three characteristics that models of legal justification must deal with are addressed successively. First, the defeasible nature of legal justification demands a logic that can handle defeasibility (section 3).

---

1 Lyrics from the 1995 album *Mellon Collie and the Infinite Sadness* by The Smashing Pumpkins.

2 For a discussion of some of these models, see Chapter 5.
Second, the open nature of law makes that legal justification cannot be modeled satisfactorily as a product, but must be modeled in an open procedural model instead (section 4). The final characteristic is one that makes it hard for any model to represent legal justification, and is addressed in discussing the Münchhausen Trilemma (section 5). A procedural alternative to model justification, i.e., a dialogical model, is proposed in section 6; in the subsequent section it is indicated how the characteristics can be dealt with in a dialogical model. Next the theoretically interesting issue how to justify dialog rules is discussed. Finally, it is discussed whether the dialogical procedure is either a perfect, imperfect or pure procedure [Rawls, 1972]. Because the dialogical procedure is viewed as none of these three procedural types, an alternative named 'the legal procedure' is proposed.

1. THE PRODUCT AND THE PROCESS OF JUSTIFICATION

There is a difference between the product of justification and the process of justification. Both aspects of justification can be recognized in Collins Dictionary's description of justification:

- reasonable grounds for complaint, defence, etc.;
- the act of justifying.

The first description deals with the product of justification. If the product of justification is studied, general structures of support or justification between sets of premises and conclusions are defined. In a product approach of justification a statement is justified:

- if the premises are justified, and
- if by valid inference
- the conclusion is derived from the premises.

The product of argumentation is static.

```
REASONABLE
GROUNDS

Justify

STATEMENT

THE PRODUCT OF JUSTIFICATION
```

Assume the statement must be justified that Jeff hit a red light and did not stop. A reasonable ground could be that a police officer saw him doing it, and that testimony by police officers is assumed to be true. An example of an unreasonable ground could be the sole fact that Jeff is someone who regularly runs red lights, and that therefore it can be assumed he has done so this time too. The circumstantial evidence based on Jeff's behavior in the past, will not be sufficient to justify the claim about Jeff's offence.

The difficulty with the product of justification is to draw the line between reasonable and unreasonable grounds.
The second description of Collins Dictionary deals with justification as a process. The process of justification is the exchange of information that is introduced step by step in order to justify a statement; each step corresponds to a stage of the process in which a statement is either justified or not. If the process of justification is studied procedural rules are defined that determine for each stage of the process whether a statement is justified.

In a process approach of justification a statement is justified.

- if after a sequence of one or more steps, the statement is justified according to the rules of the procedure.\textsuperscript{3}

The process is not static, but dynamic. The act of justifying is successful if the statement is justified in the final stage.

---

Let us consider Jeff's case again. Statements are put forward one by one to justify that Jeff ran a red light. These adduced statements can be acceptable (e.g., the policeman's observation) or not. The process continues until the final result is satisfactory.

The difficulty with the process of justification is to define procedural rules that guarantee that the procedure is a justifying one.

The distinction between justification as a product and as a process is important. In this chapter I will first concentrate on the product of justification, and second on the process.

So far the discussion of justification was not particularly legal. In law justification plays an important role,\textsuperscript{4} and maybe most prominently in judicial decisions. Judicial decisions often encroach deeply on social life. Those concerned with a verdict (either directly or indirectly) want to know why the judge decided the way he did. This is the main reason why a judge has to state grounds for his verdict.

In the Netherlands the constitution prescribes that legal verdicts must contain the grounds on which they are based.\textsuperscript{5} In criminal law we can find a similar demand, e.g., that

---

\textsuperscript{3} If the process is a dialog, a legal statement is justified if after a sequence of one or more speech acts, the audience is convinced of the justice of a legal statement.

\textsuperscript{4} Justification is not only important in law, but also plays, for example, a major role in ethics [e.g., Stevenson, 1944; Wellman, 1971] and epistemology [e.g., Pollock, 1974].

\textsuperscript{5} See section 121 of the Dutch Constitution (art. 121 Grondwet): "...houden de vonnissen de gronden in waarop zij rusten."
the verdict has to mention the factors that determined the penalty. In civil law the verdict must contain the grounds for the decision, concerning both legal and factual matters.

2. Justification as a Product

The deductive model is probably the best-known way in which justification can be modeled as a product. Legal scholars are traditionally attracted to a deductive model of justification. Not surprisingly, since the deductive model is both simple and powerful.

In general, deduction can be defined as follows: if the truth of a set of sentences, called the premises, guarantees the truth of a sentence, called the conclusion, then the derivation of the conclusion from the premises is deductively valid. In terms of justification a sentence is justified if it is derivable in a deductively valid way from a set of justified premises. So there are two conditions that must be satisfied before a sentence is justified:

1. it must be possible to derive the sentence in a deductively valid way from the premises, that is, the derivation must be based on a deductively valid argument;
2. the premises must be justified.

Propositional Logic will be used to show some deductively valid arguments. An example of a valid argument is:

\[
\begin{align*}
A \lor B \\
\neg A \\
\hline
\text{therefore } B
\end{align*}
\]

As a model of justification, this example reads as follows. The first premise means that \( A \lor B \) is justified, the second premise that \( \neg A \) is justified. These two premises justify the conclusion \( B \).

A second example of a valid deductive argument is the reasoning scheme known as Modus Ponens.

\[
\begin{align*}
A \rightarrow B \\
A \\
\hline
\text{therefore } B
\end{align*}
\]

The first premise, \( A \rightarrow B \) (the material implication), can be read as “if \( A \) then \( B \).” This premise is due to the popularity of Modus Ponens in the law: the material implication is identified with a rule, and rules, particularly the application of rules, play an important role

---

6 See section 359-5 of Dutch Code of Criminal Procedure (art. 359, lid 5 Sv): “Het vonnis geeft in het bijzonder de redenen op, die de straf hebben bepaald.”

7 See section 59-1 of Dutch Code of Civil Procedure (art. 59 lid 1 aanhef en sub 3 Rv): “Het vonnis (...) moet behelzen de gronden der uitspraak, zoo wat de daadzaken als het regtspunt, ieder afzonderlijk betreft (...)”

8 Besides the deductive model, there are many other product models. For instance, most nonmonotonic logics model the product of reasoning instead of the process.
in legal reasoning. The Modus Ponens argument runs as follows: if $A \rightarrow B$ and $A$ are justified, then $B$ is justified. A legal example of the Modus Ponens argument is taken from Dutch Family law: if a woman is pregnant of a child, and this child is born dead, the child is regarded never to have existed. In the following argument, the conclusion \textit{Never\_existed} is justified by the two premises.

\[
\begin{align*}
\text{Born\_dead} & \rightarrow \text{Never\_existed} \\
\text{Born\_dead} \\
\hline
\text{Therefore Never\_existed}
\end{align*}
\]

The argument schemes deal only with the structure (syntax) of reasoning. However, if an argument is deductively valid, this does not guarantee that its conclusion is justified. Recall that in order to justify, the premises had to be justified too. Consider for example the following argument.

\[
\begin{align*}
\text{Born\_dead} & \rightarrow \text{Adult\_at\_18} \\
\text{Born\_dead} \\
\hline
\text{Therefore Adult\_at\_18}
\end{align*}
\]

The strange conclusion – that someone of whom we know that he was born dead becomes an adult at the age of eighteen – is caused by the false first premise. Derivations like these cannot be prevented by only checking the syntax. Therefore the characterization of justification also includes the condition that the premises have to be justified. In other words, a deductively valid argument provides a justification of its conclusion if and only if all of its premises are justified. The demand for justified premises is further elaborated in section 5. First, two other characteristics of legal justification are discussed: the defeasible nature of legal justification (section 3) and the law’s open nature (section 4). The discussion of these characteristics leads to constraints for a model of legal justification. One form of a deductive argument, Modus Ponens, is used to illustrate these two characteristics.

In the context of the law the Modus Ponens argument can be interpreted as follows: a legal conclusion ($B$) is justified by a (valid) legal norm ($A \rightarrow B$), and a fact that satisfies the condition of the norm ($A$). I will refer to this model of legal justification as the \textit{subsumption model}. Although the \textit{subsumption model} is rarely accepted as a satisfactory model of legal justification, it serves as an initial model of legal justification. The subsumption model is used to illustrate the drawbacks of deductive models in general.

3. \textbf{THE DEFEASIBLE NATURE OF LEGAL JUSTIFICATION}

Since rules are characterized by their general nature, there is often a need for exceptions in special cases. For instance, there are many exceptions to the rule that someone who kills another person should be punished. As long as these exceptions do not apply, the conclusion of the general rule can be derived provisionally. If an exception applies, the initial conclusion must be withdrawn.

\[9\text{ Section 2 of the first book of the Dutch Civil Code (art. 1:2 BW)}\]
This phenomenon can be described as the defeasible nature of legal reasoning. By defeasibility is meant that if additional information is taken into account, the status of a conclusion can change. By knowledge of additional facts, rules, etc., a justified conclusion can become an unjustified one. The initial conclusion is as it were defeated by this additional information.

There are basically two causes of defeasibility. In the law these causes are represented by exceptions to rules and by conflicting rules.\textsuperscript{10}

I will first give examples of legal arguments in which defeat is based on exceptions to a norm and conflict of norms respectively. By means of these examples it is shown that defeasibility cannot be adequately modeled in the subsumption model.

### 3.1 Exceptions to rules

Sometimes conclusions contrary to the wording of the norm are desired. In these cases straightforward application of the norm would lead to an unacceptable conclusion. Let the following example based on Fuller [1958] illustrate this. There is a norm that forbids people to sleep in railway stations. Suppose that at the end of the day Lon is waiting for a train and accidentally falls asleep. Most people, including the railroad police, will not easily apply the norm to Lon. However, applying the norm automatically would lead to a fine for Lon. Is it possible that Lon will not be fined?

- \( \text{sleeping} \rightarrow \text{fine} \)
- \( \text{sleeping} \)
- \( \neg \text{fine} \)

The above argument is obviously not a valid argument, but represents what is desired in this specific situation.

In the subsumption model, actually in any deductive model, a conclusion that is justified remains justified irrespective of the addition of any new information. The fact that Lon did sleep (in combination with the norm) inevitably leads to the conclusion that he should be fined. Once derived, this conclusion cannot change.

A way to avoid deriving that Lon should be fined is to restate the rule. For instance, by narrowing the rule's condition in a way that Lon's sleeping does not satisfy the condition anymore. The rule could become: \( \text{sleeping} \land \text{sleeping\_disturbs} \rightarrow \text{fine} \). Because Lon's sleeping was not of the disturbing kind, the condition of the rule is no longer satisfied and as a consequence the conclusion that Lon must be fined no longer follows.

Although in this way the subsumption model is capable of dealing with the problem of defeasibility, there is an objection against reformulation of rules. Since there will be more situations in which a conclusion is not desired, it could be the case that several versions of

\textsuperscript{10} The argument representing the exception to a rule is usually referred to as an undercutting argument, the argument representing conflicting rules as a rebutting argument. The terms undercutter and rebuttter stem from Pollock [1987].
the same rule are necessary, or that after a number of reformulations the original rule becomes unrecognizable.

A better way to deal with this problem is to model the rule and the exception separately. In that way exceptions can be modeled without the need to reformulate the rule. This approach is impossible if a deductive logic is used like is in the subsumption model, because the information that an exception to a rule applies cannot change the conclusion based on the rule.  

3.2 Conflicting norms

Legal norms can conflict. It is hard, if not impossible, to design a conflict free system of norms. There are even lawbook sections in which the legislator explicitly admits the possibility of a conflict. For instance, someone is not punishable if a legal norm forced him to behave criminally. So someone can be forced by a legal norm to break another legal norm. The application of two conflicting norms can be represented in the subsumption model as follows.

\[ A \rightarrow B \quad C \rightarrow \neg B \]
\[ A \land C \quad C \land A \]
\[ \text{therefore } B \quad \text{therefore } \neg B \]

The two arguments represent two applicable conflicting norms. As a consequence, both B and \( \neg B \) are justified. The phenomenon of incompatible conclusions lies at the heart of law: in many legal cases two opposite conclusions are defensible. In the subsumption model, opposite conclusions cause a problem, because a consequence of contradiction is that anything can be derived.

A way to prevent contradiction is to reformulate rules. Suppose that thedrafter of the rule has indicated to prefer one rule above another. For instance, the second rule could be reformulated as: \( C \land \neg A \rightarrow \neg B \). This means that only if A is not the case, the conclusion \( \neg B \) can be justified. Since both A and C are the case, \( \neg B \) cannot be justified after the reformulation of the rule. There is no conflict left and only B is justified.

\[ A \rightarrow B \quad C \land \neg A \rightarrow \neg B \]
\[ A \land C \quad C \land A \]
\[ \text{therefore } B \quad \text{???} \]

---

11 How exceptions are modeled in DiaLaw, see Chapter 3, section 5f. Prakken [1993 & 1997, Chapter 5] gives an overview of techniques to model exceptions.

12 In nonmonotonic logics conclusions can change. There are numerous so-called nonmonotonic logics of which Reiter’s [1980] Default Logic is probably the best known. In a survey of these logics Ginsberg [1994, p. 2] describes nonmonotonicity as: “On learning a new fact (...) you were forced to retract your conclusion (...”).

13 See section 42 of the Dutch Penal Code (art. 42 Sr): “Niet strafbaar is hij die een feit begaat ter uitvoering van een wettelijk voorschrift.”
For three reasons this style of dealing with rule conflicts is not ideal. First, there may be more exceptions, so the condition representing all exceptions can become extremely long. A second objection is that the negative condition (and from the first objection it follows that this condition can be rather long) must be known in order to derive the conclusion. Finally, a representation with negative conditions differs from how lawyers actually reason. A conclusion is not justified after considering all possible exceptions. Only if exceptions are known or become known to be present, they are taken into account. Before that, a conclusion is justified provisionally.

A better alternative is to leave the rules in their original state, and to formulate explicitly a prioritization for cases in which the arguments lead to incompatible conclusions. This is impossible in the subsumption model, because it would only lead to more contradictions.

4. **The open nature of law**

A system may be said to be open if new elements can be added to it that were not a part of the system before. Law can change during a justification. This is why the law is said to be an open system. As the Dutch legal theorist Scholten [1974, p. 76] puts it: "(...) a system that because of its nature neither is nor ever can be completed, because this system is the foundation of decisions, while at the same time these decisions add something new to the system. I think it should be called an open system."  

I will first give two examples of law’s open nature: existence of cases without an applicable norm, and the vagueness of legal language. Along the way I will show that although some aspects of the law’s open nature can be modeled in the subsumption model, not all aspects can.

4.1 **Cases without an applicable norm**

After a surgical intervention a doctor forgot to put back the IUCD (coil) of his patient. Because of this mistake the woman became pregnant. Five years before the doctor’s mistake, she had decided that two children were all she and her husband could take care of financially. She sued the doctor. Her claim for damages primarily concerned costs for loss of income and costs for education.

When the Dutch Supreme Court had to decide this issue in 1997, 15 there had been no precedent, and legally there were no regulations about such claims. The court had to create new law and decided that the doctor had to pay for the costs the woman claimed. The opinions about this decision were divided. Some were not convinced about the ethical correctness of the decision [e.g., Leijten, 1997], others considered the decision a good addition to the system of law [e.g., Hirsch-Ballin, 1997].

---

14 Originally in Dutch: "(...) een systeem dat uit zijn aard niet af is en niet af kan zijn, omdat het grondslag is van beslissingen, die aan het systeem zelf iets nieuws toevoegen. Ik meen, dat dit het beste uitkomt, indien we van een open systeem spreken."

The example of a case in which there is no applicable norm can be modeled in the subsumption model as follows.

\[ \text{A} \rightarrow \text{B} \]

Given fact A the conclusion B is desired, but there is not an applicable legal rule. By adding the premise \( A \rightarrow B \), representing a newly created rule, the scheme can be made deductively valid.

The argument that results from adding the rule can be modeled as Modus Ponens. It is, however, not possible to model the addition of the rule in the subsumption model. That is because the subsumption model represents the static product of justification. The addition itself is an act that can only be modeled if justification is represented as a process. So, although it is possible to represent the results of the addition of a rule in the subsumption model, this representation does not fully capture the law's open nature.

4.2 Vague legal language

Legal language is vague, and it is often not immediately clear whether the condition of a legal rule is satisfied. The following case is an example of a vague legal term that was interpreted using a newly created interpretation. Therefore, the decision added to the system of law.

A group of actors called Cicero violated the copyright concerning the piece they were playing. Although the violation was not entirely intentionally, the Dutch Supreme Court decided that intentionally also meant: "knowingly and willingly accepting the ... chance that ...".\(^{16}\) This still widely used interpretation of the term 'intent' was added to the law by the above decision. As a consequence of this decision a greater amount of criminal acts can be qualified as intentionally.

In general, terms that are in need of interpretation give the opportunity to justify a statement using new interpretations that add to the system of law. In terms of the subsumption model, vagueness can be represented as follows.

\[ \text{A} \rightarrow \text{B} \]

\[ \text{C} \]

\[ \text{therefore} \quad ??? \]

In order to obtain a deductive argument the gap between C (the fact) and A (the condition of the rule) must be bridged. There are two ways this can be done: either argue that C is a case of A, or argue that A embraces cases like C. In law these approaches are commonly referred to as classification and interpretation respectively. Interpretation concerns the rule, and classification concerns the facts. However, interpretation and classification supplement

\(^{16}\) Decision on November 9, 1954 (NJ 1955, 55).
each other without there being a clear border. In the present example a rule of interpretation (or classification) can be added in order to construct a deductively valid argument: \( C \rightarrow A \).

As in the case of the addition of a new rule, the form of argument that results from adding the interpretative premise adequately represents the logical structure of the resulting argument. However, it does not represent the addition of this interpretive premise to the system. The subsumption model only has an eye for legal arguments as a product, not for their construction by adding to the law and exploiting the law’s open nature. In this respect, the subsumption model is not suitable to account for legal justification.

In order to model adequately the law’s open nature, a procedural model is necessary that allows the addition of new facts and rules.

It was argued that conclusions can be justified by arguments based on newly created rules. Obviously, this also holds for conclusions that result from an exception to some rule. So exceptions can also be based on new law. A consequence is that each rule may be excluded by a newly created exception. Does this imply that rules are useless? Not at all, rules are a good starting point, and often a justification can be based on the application of a rule alone. However, just as statements may be justified by unanticipated arguments, there are cases in which arguments can be excluded by unexpected exceptions.

5. The Münchhausen Trilemma

In section 2 it was hypothesized that "a deductively valid argument provides a justification of its conclusion if all of its premises are justified". This means that an argument must be based on justified premises. The demand for justified premises is difficult to meet. Especially in the law, which is an open system, justification can theoretically go on forever, because there is no natural ground to base a justification upon.

Statement

\[ \overset{\text{justifies}}{\overline{\text{Premise}}} \]

\[ \overset{\text{justifies}}{\text{............}} \]

\[ \text{no natural ground} \]

A lawyer’s justification may be compared to a mathematical proof. There is, however, an important difference. A solid mathematical proof is universally true, a proof “stands once and for all” [Loui, 1992]. A legal justification can never be that solid, since the system of law is not based on axioms.\(^{17}\) Once premises are gathered that justify a legal statement, it still remains an open question whether those premises themselves are justified. Because of the recursive nature of justification—the premises that justify a statement are statements themselves that need to be justified— it seems impossible to obtain truly justified premises.

\(^{17}\) Nevertheless Rödig [1980] has tried to axiomatize the law [cf. Gordon, 1995, p. 35ff.]
The search for justified premises leads to what Albert [1975] called the *Münchhausen Trilemma*. The three branches of the trilemma are:

1. infinite regress – justification is never ending;
2. logical circle – a statement that is being justified is used to justify itself;\(^{19}\)
3. dogmas – some statements are assumed to be justified by definition.

The first branch of the trilemma deals with the demand that each premise must in turn be justified.

\[(i) \quad I \text{ am King}\]
\[(ii) \quad \text{Because the first born child of a King becomes King}\]
\[(iii) \quad \text{Because the constitution says so}\]
\[(iv) \quad \text{Because the majority of the Parliament wanted it that way}\]
\[(...) \quad \text{Because ...}\]
\[(n) \quad \text{Because ...}\]

This on-going justification is called infinite regress. Each time a statement has been justified by another statement, the latter statement needs to be justified itself. Because the regress is infinite, there is no ultimate (and justified) premise on which a justification can be based.

The regress can be stopped by adopting one of the other two branches: a logical circle or a dogma. An example of a logical circle is the following.

\[(i) \quad I \text{ am King}\]
\[(ii) \quad \text{Because I wear a crown}\]
\[(iii) \quad \text{Because I am King}\]

In case of a logical circle one cannot speak of a true justification. If it were considered a justification, then every statement could be justified by itself. From a different point of view the circle does not even stop the regress, namely if the circle is seen as an infinite loop.

The last branch of the trilemma deals with founding the justification on grounded statements.

\[(i) \quad I \text{ am King}\]
\[(ii) \quad \text{Because the first born child of a King becomes King}\]
\[(iii) \quad \text{Because the constitution says so}\]

In the infinite regress example the justification continued at this point. However, it is possible to stop the justification at a certain point. A possible reason to stop could be that it

\(^{18}\) Albert [1975, p. 13]: “Das führt zu einer Situation mit drei Alternativen, die alle drei unakzeptabel erscheinen, also: zu einem Trilemma, das ich (...) das Münchhausen Trilemma nennen möchte.”

The reason Albert names it after the Baron von Münchhausen is because of the parallel between this trilemma and a problem the Baron once faced when he fell into a swamp. What the Baron needed was a basis to lift himself up, just as in this trilemma a solid basis to ground an argument on is needed. The Baron solved the problem by pulling himself out of the swamp by his own hair.

\(^{19}\) Also known as petitio principii.
is generally accepted to use the statement as an ultimate justification. In this case it means that a statement referring to the constitution does not have to be justified. Such grounded statements are dogmas, comparable to mathematical axioms. They are called dogmas, because it is not deemed necessary to justify these statements. However, it is not the case that they never need to be justified. Sometimes it is necessary to allow arguments against these legal dogmas. The following example illustrates this.

The literal wording of a statute is often the starting point for a justification. However, there can be arguments against the literal wording of a statute. In the Netherlands, at its introduction legislation is published in the Netherlands Bulletin of Acts and Decrees and the Netherlands Government Gazette, and after this only the amendments to the legislation are published. An up-to-date, complete version of the statute is only available through commercial publishers. Because of the complexity of the amendment regulations, it happens that publishers make mistakes. For such occasions, arguments that propose a different text than the literal wording of the statute have to be allowed. These arguments would not be possible if the literal wording of the statute was considered a dogma.

6. JUSTIFICATION AS A PROCESS: A DIALOGICAL MODEL

In the last three sections we encountered three characteristics of legal justification.

First, the defeasibility of legal reasoning asked for a model that can deal with both conflicting rules and exceptions, without it being necessary to reformulate the rules.

Second, the open nature of the law asked for a process model in which new law can be introduced. So, an open procedural model is required.

The Münchhausen Trilemma pointed out the unpleasant consequences of the demand that conclusions should be based on justified premises. Because of the law’s open nature, justification can lead to an infinite regress. Two ways to end this regress are by logical circles and indisputable premises (dogmas). None of the three branches of the trilemma were considered acceptable.

In the next section I will indicate how a dialogical model deals with the three characteristics of legal justification (defeasibility, open nature and the Münchhausen Trilemma). In the

---

20 An example from legal theory of an ultimate statement that needs no further justification is the ‘Grundnorm’ of Kelsen [1960]: Hart’s rule of recognition needs no justification either: “For whereas a subordinate rule of a system may be valid and in that sense ‘exist’ even if it is generally disregarded, the rule of recognition exists only as a complex, but normally concordant, practice of the courts, officials, and private persons in identifying the law by reference to certain criteria. Its existence is a matter of fact.” [Hart, 1960, p. 107; italics are mine]

21 Smith [1994, p. 29] argues that the Münchhausen Trilemma is not really a trilemma, because a justification stops at the point a premise is generally accepted: “at a certain point it is no longer rational to ask for further justification”. However, this does not solve the Trilemma, it only shifts the problem. It still is not clear under what circumstances it is no longer rational to ask for further justification.

22 In Dutch: Staatsblad en Staatscourant.
From law to DiaLaw - Why legal justification should be modeled dialogically

remainder of this section a procedural model, i.e., a dialogical model of legal justification is proposed. But first some relevant previous work is addressed briefly.

Perelman's rhetorical theory [Perelman & Olbrechts-Tyteca, 1971] and Habermas' [1973] consensus theory of truth have influenced most of the current dialogical research. Today's best-known defenders of dialogical models in law are Aarnio [e.g., 1987], Alexy [e.g., 1989] and Peczenik [e.g., 1989]. In the beginning of the eighties they integrated their theories that were developed independently in the seventies [Aarnio, Alexy & Peczenik, 1981]. On an abstract level, their approach is similar to the one that will be proposed here.23

The Erlangen School [Lorenzen, 1969; Schwemmer, 1971] defined a procedure by which a system of ethical norms can be constructed. This procedure was modeled as a discussion in daily life language. An interesting element of their procedure is the idea of consensus: assumptions are determined on the basis of acceptance by others. This idea also lies at the basis of the proposed procedural model of justification: statements are justified if they are accepted.

The procedural model of justification does not concentrate on the specific structure of reasoning schemes (the product of justification). Instead, the focus is moved to the procedure in which statements are justified. In the field of law this is not an uncommon method. For instance, legislation originates after a particular legislative procedure. The content of a statute is at the most of marginal interest for the validity of a legal norm. If a majority of the Parliament votes for a certain statute, this statute becomes part of the law. Of course citizens can have objections about the content of norms, but they can only raise their objections in another procedure: elections.

Basically, a dialogical model of justification can be characterized as follows:24

- there are at least two participants;
- rules regulate the exchange of arguments between these participants;
- statements are justified if the participants agree.

It was recognized long ago by Protagoras that "each case has two sides" [cf. Nieuwenhuis, 1992, p. 98]. Therefore, there have to be at least two parties, representing each a side of the dispute. The dialog model proposed here has exactly two parties. However, two parties do not necessarily mean that two persons are involved. Even a single person's attempt to justify a statement can be modeled as a two-person dialog game in which the person alternately plays the role of the one who attacks the statement and the one who defends it. Or, according to Barth & Krabbe [1982, p. 12]: "Reasoning as carried out by one person should be studied as an (important) special case, viz. the case where the two parties coincide in one person: the self-critical case." Of course, opinions of others can play a role in such a one-person dialog.

---

23 See for example Aarnio [1987, p. 185-187]: "Justification is a procedure of rational practical discourse (...) justification procedure is essentially a dialog."

24 The discussion of the dialog game is rather abstract. In more detail the elements of a dialog game, DiaLaw, are discussed in the next chapter. For now the precise details are not important, because the main line suffices to make my point.
There must be a set of rules that defines when parties are allowed to adduce statements, arguments, etc. The rules guide the procedure, comparable to a game that is played according to rules.

Only within the dialog game statements can become justified. The justification is not related to some independent criterion outside the procedure. Instead, justification is defined relative to the parties of the dialog game. Only if the parties want to hear reasons for a statement these reasons have to be adduced, and only if a statement is accepted, it is justified.

The dialog game is a rhetorical procedure [Witteveen, 1988]. Characteristic for such a procedure is that there is no predetermined outcome, the procedure is non-deterministic. By presenting reasons, each party tries to draw the outcome in his direction, but the final result cannot be determined in advance.

The outcome of a dialog only holds for the participants at the moment they finish their dialog. The justification is relative to a particular audience and relative in time. In a new game the outcome of a previous game can be discussed. Such a discussion is comparable to, for example, the evaluation of a court decision by legal scholars.

7. DEALING WITH DEFEASIBILITY, OPEN NATURE, AND THE MÜNCHHAUSEN TRILEMMA IN A DIALOGICAL MODEL

For each of the three characteristics I indicate how the dialogical model deals with them: the defeasible nature of legal justification, the open nature of law, the Münchhausen Trilemma.

Defeasible nature of legal justification
In the dialog game a variant of Reason-Based Logic (see Chapter 3 and [Hage & Verheij, 1994]) is used. Both counterarguments and exceptions can be modeled. Since exceptions can be formulated separately it is not necessary to reformulate rules. Moreover, application of a rule only gives rise to a reason for the conclusion, the conclusion itself does not follow directly. This makes it possible to derive a conclusion opposite to what strict rule application would lead to.

In a dialog game defeat can occur in several ways. For instance, a party claimed a statement and because of an exception adduced by his opponent his initial belief that the statement is justified is defeated. As a consequence he retracts his statement. Assume a party claims that Bob must be punished because he shot the sheriff, and his opponent convinces him that Bob acted in self-defense. In that case his initial conclusion is defeated, and he will withdraw it.

25 Previously, the non-deterministic character of law has been used as an argument against logical models of legal reasoning. For instance, Berman & Hafner [1987] and Moles [1992] attacked logical models, comparable to the subsumption model. To quote Moles: "The latter's [judge] role involves what may be called a 'performative utterance' which is much more a matter of rule creation than it is of rule following. This (...) is a factor which logical modelling cannot account for."
There are other situations conceivable, but the example is sufficient to illustrate how a dialogical model can deal with defeasibility. Namely, if a statement is claimed, the party who claims it believes that the statement is justified. Because of later information (the temporal aspect) put forward in the dialog, the statement can become defeated. In that case the party who claimed the statement will (be forced to) retract his claim.

**Open nature of law**

The participants of the dialog game are not restricted to a fixed set of arguments, statements or premises. In the dialog game new facts, rules, reasons, etc., can be introduced freely. Reasons that are adduced to support statements do not have to be based on the application of an existing rule. Each reason, if accepted, is allowed as a justification of a statement.

**The Münchhausen Trilemma**

No branch of the Münchhausen Trilemma was satisfactory. Alexy [1989, p. 179] sees the definition of a procedure as a way to overcome infinite regress: "...dropping the demand for ever further justification of every statement by another statement, in favor of a set of requirements governing the procedure of justification." Above, justification was defined relative to the parties of the dialog game. Does this guarantee that the problems of the trilemma are solved?

In order to prevent circles it will not be allowed that a statement is used to justify itself. In a way dogmas and regress are each other’s opposites. On the one hand, if dogmas are not used, it is allowed to question each statement; if each statement can be questioned, one lands easily in a situation of infinite regress. On the other hand, if one does not want that infinite regress occurs, then the justification must stop at a certain point. If termination points are defined outside the procedure, they can only be regarded as a dogma. However, without defining dogmas it is still possible to stop infinite regress. In the dialog game the justification can stop once the other party accepts a statement. It could be said that this accepted statement is a dogma in the discussion. The difference with real dogmas is that there always is an opportunity to question a statement, which is not allowed in case of a dogma. Note, however, that although regress can be stopped without using dogmas, it cannot be guaranteed that the regress in fact stops.

**8. Justification of dialog rules**

In the next chapter the dialog game DiaLaw is defined. This dialog game is a procedural model of justification. How can it be determined whether the rules of the dialog game are suited to model the process of justification? There are various ways to justify rules of a dialog game or discourse rules [Alexy 1989, p. 180ff.] For instance, rules can be formulated to obtain a desirable goal. If this goal is achieved by these rules, the discourse rules can be called justified. A weak point here is that the justification is only relative to the goal. Moreover, the problem is shifted to the goal, that may need to be justified itself.

If discourse rules are presented without adducing any reasons for their usefulness, except that the set of rules describes the practice of discussions, the justification is called definitional [Alexy 1989, p. 184]. Although this can hardly be called a justification of the dialog rules, it seems that most sets of discourse rules are 'justified' in this way.
No matter whether and how dialog rules are justified, I believe that in the end each set of discourse rules always remains an arbitrary set, put together on the basis of an individual choice of the designer. Therefore it is important to allow discussion about the discourse rules. Without a possible discussion about the rules, one lands in one of the branches of the Münchhausen Trilemma, namely that a justification is based on dogmatic dialog rules. It may seem strange that the rules of discussion become subject of a discussion that is guided by the same rules. But these types of discussions do exist. For instance, the discussion rules of a meeting can be altered during the meeting.

Although justification of dialog rules is theoretically of interest for a dialog game, the realization in an actual working system is hard. In DiaLaw discussion of the dialog rules is yet not realized, but hopefully will be in the future.

It is a big challenge to make systems like DiaLaw with altering dialog rules. Whether this is practically possible or whether this remains only a theoretical desire is still an open question. I hope for the first, but I fear the second. There is, however, some progress on this point. Systems already exist with a few (not all) discussion rules that can be altered during the discussion. For instance, Vreeswijk [1995] has formalized and implemented a version of Nomic.26

9. HOW PURE IS THE PROCEDURE OF LEGAL JUSTIFICATION?

The dialogical model does not represent the product of justification, but models legal justification procedurally. Rawls [1972, p. 85f.] distinguishes perfect, imperfect, and pure procedures. In previous joint work [Hage, Leenes & Lodder, 1994; Leenes, Lodder & Hage, 1994] the procedural view of law was put into perspective using the terminology of Rawls; we qualified our approach as pure procedural. In order to characterize the dialogical model of justification more accurately, I will now propose a fourth category, appropriately called the legal procedure. But first, let me explain what the three procedure types of Rawls mean.

There are two characteristics of a perfect procedure. First, it must be possible to formulate a criterion, independent from the procedure, by which can be decided what is right and what is wrong. Second, it must be possible to define a procedure that is guaranteed to lead to the desired, correct result. Earlier in this chapter, I claimed that there is no independent criterion by which it can be determined whether legal statements are justified. Therefore, legal justification cannot be viewed as a perfect procedure.

In case of an imperfect procedure it is possible to define a criterion by which can be decided what is right and what is wrong. But, in contrast with the perfect procedure, it is impossible to define a procedure that is guaranteed to lead to the desired result. Since an independent criterion cannot be defined, legal justification cannot be viewed as an imperfect procedure either.

---

26 Nomic is a game in which players can alter the rules of the game. Nomic was designed by Peter Suber in 1980 and described by Douglas Hofstadter in his "Metamagical Themas" column in Scientific American of June 1982.
In pure procedures there is no criterion by which can be decided what is right and what is wrong. There is no standard to evaluate the outcome of the procedure. You can only fight the outcome of the procedure by arguing that the procedure was not applied properly.

The following table gives an overview of the three different procedures.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Independent criterion</th>
<th>Procedure that is guaranteed to lead to desired result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perfect procedure</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Imperfect procedure</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Pure procedure</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

I already claimed that there is no independent criterion by which it can be decided whether a legal statement is justified. In case such a criterion would exist, this criterion could be used to determine whether a statement is justified, independent from a procedure. Or, in case the justification were modeled as a procedure, the outcome of the procedure could be verified as correct or not on the basis of the criterion.

The lack of an independent criterion makes it necessary to model justification as a process. The desired result of a procedure of legal justification is that the status of a statement, justified or not, is established. At the end of the procedure the statement can be either successfully justified, or the justification may have failed. In case of failure there can be:

- agreement that the statement is not justified, or
- disagreement about whether the statement is justified.

So at the end of a procedure a statement is justified, not justified, or the status of the statement is undecided. However, this describes only part of the desired result. The actual question is whether in case a statement is justified in the procedure, this statement really is just (fair, legitimate). Or, how can one be sure that a statement that is justified in a procedure, really is just? Conversely, how can one be sure that a statement that is not justified, really is unjust? And, finally, whether a statement which status is undecided, really is neither just nor unjust. I concentrate on the first question (how one can be sure that a statement that is justified in a procedure, really is just); the answer will apply analogously to the other two questions. So, the question is:

*How is it possible to devise a procedure that guarantees that a statement is justified if and only if it was justified in that procedure?*

A possible answer is given by Hage [1997b]. He distinguishes two pure procedures: a 'wrong' and a 'correct' procedure.

In the first, 'wrong' procedure the participants in the dialog are free to use or not use the legal rules which are applicable to their case. To me this freedom seems essential, because one of the aims of the procedure is to decide which rules are applicable. It is not possible to decide before the procedure starts which rules are applicable or not, this has to be decided in the procedure. If it were decided in advance, this would imply that there is some independent criterion after all. Namely, that some rules are applicable and other rules are not, and that the applicable rules have to be used.
The independent criterion is smuggled into the second and 'correct' pure procedure. Only if an independent third safeguards that the applicable legal rules are used, the outcome of the procedure can be really just.\textsuperscript{27} It is clear that in legal practice a judge is such an independent person. How this third person can be represented in a model is a theoretical question of which Hage admits it has not been answered yet. I think that the moment we can model such an independent third, we have found an independent criterion to check whether legal statements are justified.

My answer is of a different nature. There is no other way to determine whether a statement is justified than following a procedure of justification. Only in a process, a statement can become justified. This means that at each moment in time, only procedurally justified statements do exist. So it seems that the procedure of justification is pure. However, if the procedure of justification were pure, this would mean that there exists a procedure by which it can be guaranteed that all statements that are justified in the procedure are really just (fair, legitimate). I believe that it is impossible to define a procedure that guarantees that the statements justified in the procedure are really just.\textsuperscript{28}

Does this mean that the procedure of justification is imperfect after all? In case of an imperfect procedure the desired result is also not guaranteed. The difference is, however, that in case of an imperfect procedure it is not necessary to apply the procedure. The imperfect procedure is used because it is considered the best way to establish a correct result. In case of legal justification a procedure is essential. A procedure is not the best way to establish that a statement is justified, it is the only way to determine whether a statement is justified.

An additional argument against considering the procedure a pure one, is that if the procedure were pure, only reasons based on procedural shortcomings could set aside the outcome of the procedure. However, there are also arguments that are not about procedural shortcomings. For instance, grounds to set aside a verdict are often substantial, viz. non-procedural. Note, however, that also the non-procedural arguments are adduced in a procedure.

I would like to propose an alternative procedure, and appropriately call it 'the legal procedure': there is no independent criterion and applying the procedure is the only way to justify statements, but the procedure cannot guarantee that the statements are really just (fair, legitimate). Statements can be justified in a procedure only, but whether the statements are really just is a question that cannot be answered.

Summarizing, statements are justified if they are defended successfully in a dialog. A statement is justified within the procedure only if there is agreement between the participants. Comments on the outcome of a dialog game can be non-procedural, but can only be put forward in a new procedure. Whether statements that have been justified are:

\textsuperscript{27} Instead of justified statements, the outcome of the procedure in Hage's paper is law.

\textsuperscript{28} An alternative is presented by Aarnio [1987, Chapter 4] who calls the outcome of a procedure only justified, if the 'lebensform' the participants of the dialog belong to would accept the outcome. This alternative seems attractive, but the problem is how it can be determined whether the 'lebensform' would accept the outcome.
really just (fair, legitimate), is a question that cannot be answered. Ideally, in most cases they will be, and in only few they will not be. The aim is to develop a procedure that guarantees in the best possible way that justified statements are really just.

10. CONCLUSION

In this chapter several aspects of legal justification were discussed in order to justify that legal justification should be modeled dialogically. According to the introduction, the purpose of this chapter was threefold.

First, convince those that did not yet see the benefits of dialogical models. As follows from this chapter, the only way to justify is to convince others so that they accept your position and make it theirs. Only time can tell whether I succeeded.

Second, describe my motives for those that are already attached to dialogical models. Comparing arguments for a shared opinion is informative. This chapter was already the result of previous comparisons. However, it is not the ultimate defense. Comments on the present chapter can make the defense of dialogical models stronger.

Finally, the development of the model DiaLaw needed justification. I believe the present chapter provides a theoretical basis for a dialogical model of justification such as DiaLaw. The next chapter will teach us whether DiaLaw meets the criteria for a model of legal justification as they were presented in this chapter.
Chapter 3

DiaLaw

A dialog game for legal justification

It's so easy, when you know the rules
... Play the game

In the previous chapter arguments for modeling legal justification dialogically were addressed. In this chapter the legal dialog game DiaLaw is defined. The game DiaLaw builds on the theory of legal argumentation developed over the years by Hage, Leenes and Lodder [Hage, Span & Lodder, 1992; Leenes, 1993; Lodder, 1993; Hage, Leenes & Lodder, 1994; Leenes, Lodder & Hage, 1994; Lodder, 1996a, 1996b, 1997b, 1997c], and was first presented in [Lodder & Herczog, 1995].

The chapter is structured as follows. First, it is indicated what constraints for the model DiaLaw follow from the results of the previous chapter. Then, an informal description of the basic concepts of DiaLaw, and some sample dialogs are given (section 2). In section 3 follows the definition of the dialogical framework. To illustrate the core of DiaLaw, general rules for communication are presented first (section 4). Some special language elements, related to Reason-Based Logic and particularly suited for legal discussions, are introduced in section 5. In section 6 are worked out special rules for communication – legal tools and forced commitment. These additional rules are necessary in order to regulate the discussion with the special language elements. The chapter ends with a short discussion of the implementation of DiaLaw in Prolog (section 7).

Each definition of the dialogical framework and each rule of the game is explained informally. It should be possible to understand DiaLaw only by reading the informal explanations. Moreover, there are examples to facilitate the understanding of the definitions and rules.

In the next chapter extensive examples of dialogs are worked out. The current and the next chapter can be read in basically three ways. First, the chapters can be read in normal order. Second, one can start by reading the examples of the next chapter. Finally, while reading the current chapter one can switch to the next one and back again. No matter in which way the chapters are read, before consulting the next chapter, section 1 and 2 of this chapter should be read.

1. Justification in DiaLaw

In the previous chapter I argued for a dialogical model of justification. The ideas described in that chapter are the foundation for the dialogical model DiaLaw. The key idea is that justification of a statement can solely be based on agreement among the participants in a dialog. There is no criterion outside the dialog that can determine what is justified.

---

1 Lyrics from the 1980 album The Game by Queen.
The definition of justification as agreement is a consequence of the undesirability of using dogmas in legal justification. Each statement may be questioned, and no statement is justified by definition, like a dogma is. If a statement is questioned, the player that claimed the statement is given the opportunity to put forward statements to justify the questioned statement.

What is justified only holds for the participants of a particular dialog. Moreover, the statements justified in one dialog can be rejected in a new dialog. So justification is not only audience dependent, but also time dependent. Justification of a statement for all, or for all rational beings, is in my opinion impossible. Although it is maybe in contradiction with the above, there is some minimal rationality in DiaLaw the players are bound by.

First, direct contradiction in the same dialog is not allowed. Note that only obvious contradiction can be forbidden. For instance, someone cannot both have murdered and not have murdered the same person. However, often there is need for extra domain knowledge in order to determine something is a contradiction. The death of a person cannot be caused both by murder and death by negligence. That these two exclude each other can only be determined on the basis of knowledge of the criminal law. Second, it is forbidden to deduce a statement, to justify itself (circles). Finally, by special language elements a player can be forced to accept or withdraw a statement.

The following quote [Hamblin, 1970, p. 244] reflects the spirit of DiaLaw:

"The logician does not stand above and outside practical argumentation or, necessarily, pass judgment on it. He is not a judge or a court of appeal, and there is no such judge or court: he is, at best, a trained advocate. It follows that it is not the logician's particular job to declare the truth of any statement, or the validity of any argument."

The rules of a DiaLaw leave freedom to the participants: they decide what valid arguments, justified statements, and acceptable premises are. The game provides a language in order to guarantee that the players understand each other. The rules concerning what counts as an argument are loose. The purpose of the game is that the players convince each other, by whatever they are saying. So acceptance always suffices for justification. When discussing special language elements, ways to force the opponent are mentioned. However, the other player can be forced only if the players agreed at least on some points.

2. Basic Concepts of DiaLaw

DiaLaw is a two-person dialog game, in which both players make moves alternately. The goal of the game is to justify statements in a dialog. The sentences put forward by one player become justified whenever they are accepted by the other player. Consider the following discussion between Bert and Ernie.

<table>
<thead>
<tr>
<th>Bert: O.J. is a murderer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ernie: Is he?</td>
</tr>
<tr>
<td>Bert: Yes, he shot his former wife</td>
</tr>
<tr>
<td>Ernie: I think you're right, Bert</td>
</tr>
</tbody>
</table>

THE BOX DISCUSSION

---

2 This topic is related to the difference between structural and procedural arguments [Lodder, 1997a]. On types of arguments see also Chapter 6.
The example shows a simple, short dialog. In the remainder of this section it will be referred to as the BOX DISCUSSION. The following concepts of the dialog game are informally introduced:

- the participants;
- the moves of the game;
- the burden of proof;
- the role of commitment;
- the dialog rules;
- levels in the dialog.

### 2.1 The participants

DiaLaw regulates the discussion between two players. DiaLaw can be played by two (groups of) people, or even by a single person (see Chapter 2, section 6).

If the players cannot agree on a statement, there are two options. First, an independent third party decides the issue. In the law the role of this third party is performed by judges, arbiters, etc. The second option is to leave the disagreement, so agree to disagree.

In DiaLaw the role of an arbiter is not modeled. If it were, it would imply that there indeed exists an independent criterion to settle conflicts, namely the criterion the judge uses to decide.\(^3\) This would be in contradiction with my claim that such a criterion does not exist.

An unpleasant consequence of not having an arbiter is that the dialog is not guaranteed to end. However, I prefer an unfinished dialog above one that ends because of a decision based on prefixed criteria. Moreover, in case a dialog does not end, apparently the sentence is not justified in the eyes of the opponent. Since justification is defined as acceptance of a statement by the opponent, a decision by an independent third party that forces the opponent to accept is not desirable.

### 2.2 The moves of the game

The players make moves alternately. These moves contain two elements: 1) an illocutionary act with 2) a propositional content.\(^4\) The illocutionary act is one of the following four:

- claim
- question

---

\(^3\) An alternative is to let the judge decide randomly. Whenever called to decide, he tosses a coin. But that is something the players could equally do themselves. No judge would be needed for that. Moreover, since justification is based on acceptance, a toss seems not appropriate. An additional problem is how to determine at what moment a player may call the judge, e.g., immediately after the claim of a statement?

\(^4\) The terms illocutionary act and propositional content are taken from Searle [1969, p. 30]. The illocutionary act types are inspired by the work of amongst others Van Eemeren & Grootendorst [1982] and MacKenzie [1979a], and changed over the years. In early work the distinction between the illocutionary act and propositional content was not made explicitly [Hage, Span & Lodder, 1992; Lodder, 1993, 1996a]. Similar sets of illocutionary acts as presented here can be found in [Hage, Leenes & Lodder, 1994; Leenes, Lodder & Hage, 1994; Lodder & Heczeg, 1995].
c. accept
d. withdraw

The propositional content of these illocutionary acts is formed by the sentences that the acts are about. In the formal definition of moves the illocutionary act and the propositional content are separately modeled. For example, the first move of the BOX DISCUSSION between Bert and Ernie would formally be:

...claim, murderer(o)...

In the examples below, these two elements of the move are not separated, but combined in one informal sentence.

Claim
If a player claims a sentence, he expresses that he believes that this sentence is justified. In principle, a player may claim any sentence. Only in some cases the claim of a particular sentence is forbidden by the dialog rules. For instance, a player cannot claim a sentence if he just claimed the opposite. So, the third move in the following dialog is not allowed.

Bert: O.J. is a murderer
Ernie: Is he?
Bert: O.J. is not a murderer

If a player denies a sentence claimed by his opponent, this is also modeled as a claim. The propositional content of this claim is the negation of the sentence claimed by the opponent. For instance, the denial of the claim that O.J. is a murderer is modeled as the claim that O.J. is not a murderer. So, the following dialog is allowed.

Bert: O.J. is a murderer
Ernie: O.J. is not a murderer

Question
If a player questions a sentence, he asks a justification of the sentence claimed by the other. Question is neither an acceptance, nor a denial, but lies just in between these two acts. A player usually questions a sentence if he is not yet convinced. In the BOX DISCUSSION Ernie questioned the claim that O.J. is a murderer, and later became convinced. On another occasion a player may question if he is already convinced, because he wants to hear arguments for the sentence. For instance, if Bert is a well-known lawyer who claims that O.J. is a murderer, and Ernie is already convinced, he might yet want to question that O.J. is a murderer.

Accept
If a player accepts a sentence, he agrees with the sentence claimed by the other. Accepting a sentence is comparable to claiming a sentence. Both the player who accepts and the player who claims, believe that the particular sentence is justified. The difference is that a claim initiates a discussion, and an acceptance ends it.

---

5 Note that the claim of a 'sentence' in this context does not mean the claim of a legal verdict. Sentence is used as a linguistic term.
Acceptance is a reaction to a claim of the other player. In the BOX DISCUSSION Ernie accepted that O.J. is a murderer, when he said that Bert was right. An even shorter example is:

Bert: O.J. is not a murderer
Ernie: I agree

Withdraw
If a player withdraws a sentence, he retracts a sentence claimed by himself. The player will withdraw a sentence if he is no longer willing to defend it. Withdraw is the opposite of a claim: withdrawal of a sentence undoes a previous claim of that sentence. Withdraw is also similar to accept: just like after an acceptance the discussion ends. A player can withdraw a sentence immediately after it is questioned, but almost always there will be several moves between a claim and a withdrawal. Moves during which the player became convinced by counterarguments, made him realize the weakness of his own position, etc.

Bert: O.J. is a murderer
Ernie: Is he?
...several moves later
Bert: I no longer defend that O.J. is a murderer

2.3 Burden of proof
The burden of proof plays an important role in regulating legal procedures. In DiaLaw the burden of proof is simple. The player who claims has the burden to prove that the claimed sentence is justified. This means that if a player has claimed a sentence and the sentence is questioned, he must adduce other sentences that support his claim.

The role of the player who has the burden of proof is usually called the proponent. For each claimed sentence the player who claimed it is the proponent of that sentence, and the other player is the opponent.

The roles of proponent and opponent can shift during the game. This means that the player who initiated the dialog is not necessarily the proponent of all sentences. An example of a situation in which the roles change, is when a claimed sentence is denied. In the following dialog the roles of proponent and opponent change.

Bert: O.J. is a murderer
Ernie: O.J. is not a murderer

In this dialog the burden of proof shifts from Bert to Ernie. After the first move Bert had the burden of proof, after the second move Ernie has. Only if Ernie withdraws his sentence, the burden of Bert revives.

2.4 Commitment
Commitment is a central notion in the dialog. Commitment originates when a sentence is claimed or accepted. For instance, in the BOX DISCUSSION both Bert, who claimed, and Ernie, who accepted, are committed to the sentence that O.J. is a murderer.
During the dialog the commitment of the players is recorded in what is called a commitment store. In the commitment store it is exactly indicated which player is committed to what sentences.

Commitment starts when a sentence is claimed or accepted. Commitment terminates when a sentence is withdrawn. The consequence of withdrawing a sentence is that the related element of the commitment store is deleted.

Commitment of a player limits him in subsequent moves. An example of such a limitation is that a player may neither claim, nor accept a sentence, when he is committed to the negation of that sentence.

To avoid that the dialog remains an informal talk, a player has means to force his opponent to accept a sentence. This is what is called forced commitment. Forced commitment is comparable to derivation in a monological logic, and occurs when a player is forced to accept a sentence, due to the sentences he is already committed to. Assume a player is committed to a reason that supports a sentence, and there are no reasons against this sentence. In case this player is not able to put forward a reason against this sentence, he is forced to accept it.

2.5 The dialog rules

Since DiaLaw is a game, there are also rules telling how to play the game. The dialog rules define:

- which player’s turn it is;
- whether a move is allowed;
- the consequences of valid moves in terms of commitment.

The first move of a dialog is a claim by one of the players, so each dialog starts with the illocutionary act of the type claim. For example, the moment Bert in the BOX DISCUSSION claimed that O.J. is a murderer, a dialog started.

The dialog rules define for each illocutionary act what moves can follow. This means that for each possible stage of the game it is defined what moves can follow.

The dialog rules also define whose turn it is. Only in a few exceptional cases the same player moves twice consecutively. Normally, the players make moves in turn.

A dialog ends with an acceptance or a withdrawal. For instance, a dialog about O.J. continues until either Bert withdraws that O.J. is a murderer or Ernie accepts it. If neither Bert withdraws nor Ernie accepts, the dialog is unfinished.

Finally, for each illocutionary act it is defined what the consequences are for commitment. Recall that basically commitment originates after a claim and after an acceptance, and ends after withdrawal.

2.6 Levels in the dialog

To structure the argumentation the dialog has levels. The initial level is 0. The dialog turns to a deeper level only after questioning. So, after Ernie questioned ‘O.J. is a murderer’, the level becomes 1. On this new level, sentences are adduced that are arguments for or against

---

6 The idea of using such commitment stores is Hamblin’s [1970], the term commitment store derives from MacKenzie [1979a].
the sentence on the previous level. If a sentence is accepted or withdrawn, the dialog returns to the level on which this sentence was claimed.

Bert: O.J. is a murderer (level 0)
Ernie: Is he? (level 0)
Bert: He shot his former wife (level 1)
Ernie: Did he? (level 1)
Bert: He stood next to the dead body with a smoking gun (level 2)
Ernie: Must have shot his wife (level 2)
Bert: Who takes away someone else’s life on purpose is a murderer (level 2)

etc.

In this dialog Bert adduces on level 1 the claim that the former wife of O.J. was shot. This is an argument for the 0-level sentence that O.J. is a murderer. Ernie questions this sentence. The argument that he stood next to the dead body supports the sentence on level 1. For Ernie this support is sufficient, so he now accepts that O.J. shot his former wife. He is, however, not yet convinced that O.J. is a murderer, so Bert continues the dialog by adducing a second argument on level 1, namely that those who kill on purpose are murderers.

This dialog will continue until Bert withdraws that O.J. is a murderer, or Ernie accepts it. An example of a finished dialog is the level-1 dialog about the claim that O.J. shot his former wife. The moment Ernie accepted this ("must have shot his wife"), the level-1 dialog stopped.

3. DIA LAw’s DIALOGICAL FRAMEWORK

In this section the dialogical framework is defined. These definitions are comparable to the definition of a playing board and the pieces of a game. How to play the game is not worked out here, but in subsequent sections about the rules of the game. Subsequently will be defined:

- the language;
- the dialog move;
- the commitment store;
- the dialog;
- the dialog tree.

3.1 The Language

The players in DiaLaw try to justify statements by claiming sentences. These sentences are expressed in the language of DiaLaw. This language is not a natural language, but the formal language of First Order Predicate Logic.

Definition 1 - the language

In the language Language the following symbols are used:

- Predicate and Function symbols - finite strings of characters, numerals and/or underscores, starting with a lower case, with an arity n (n≥0);
- Variable symbols - finite strings of characters, numerals and/or underscores, starting with an upper case;
- ( ) [ ] ^ v , ~
Formulas and terms are defined in the usual recursive way [e.g., Van Dalen, 1980]. The propositional content of an illocutionary act is always a formula. The following are examples of formulas. For each formula it is explained what symbols are used.

1. \texttt{oj\_is\_a\_murderer}
   - \texttt{oj\_is\_a\_murderer} is a predicate symbol with arity zero.

2. \texttt{is\_murderer(oj)}
   - \texttt{is\_murderer} is a predicate symbol with arity one;
   - \texttt{oj} is a function symbol with arity zero; \texttt{oj} is a term.

3. \texttt{has\_murdered(Someone, the\_ex\_of(oj))}
   - \texttt{has\_murdered} is a predicate symbol with arity two;
   - \texttt{Someone} is a variable symbol; \texttt{Someone} is a term;
   - \texttt{the\_ex\_of(oj)} is a term;
   - \texttt{the\_ex\_of} is a function symbol with arity one;
   - \texttt{oj} is a function symbol with arity zero.

The same symbol can be used as a predicate symbol and a function symbol. For instance, consider the following two formulas.

1. \texttt{oj\_is\_a\_murderer}
2. \texttt{not\_sure(oj\_is\_a\_murderer)}

The symbol \texttt{oj\_is\_a\_murderer} is in the first example a predicate symbol and in the second example a function symbol. There is an obvious connection between the two symbols: they deal with the same state of affairs, namely that O.J. is a murderer. In the first formula this state of affair is presented as if it is justified. In the second formula the uncertainty about whether the state of affairs obtains is expressed.

\textbf{Notation remarks}

I use the following notational conventions. If a variable \( v \) is used, then \( v \neq v' \), unless indicated otherwise. If the value of a variable is irrelevant, this is indicated by a raised underscore: \( \_ \). If a sentence is \( S \), the negation is \( \neg S \). A double negation of a sentence (\( \neg \neg S \)) is considered to be the sentence itself, so \( \neg \neg S = S \).

\textbf{3.2 The dialog move}

The central notion in the dialog is a move. In a move a player performs an illocutionary act concerning some sentence. The move takes place on a particular level. Therefore a move is represented by a term \( P \) for the player, a term \( A \) for the illocutionary act, a term \( S \) for the sentence, and a term \( L \) for the level. The level does not give sufficient information for an efficient representation (see section 3.5). Therefore an element \( B \) is needed that indicates either for which sentence the move is an argument, or to which sentence the move is a reaction. This leads to the following definition.
Definition 2 - the dialog move

A dialog move $M_i (i > 0)$ is an ordered 5-tuple ($P$, $A$, $S$, $B$, $L$), where
$P \in \{\text{Player1, Player2}\},$
$A \in \{\text{claim, question, accept, withdraw}\},$
$S, B$ are formulas of Language,
$L \in N$ (set of natural numbers, including zero).

The value of the variable $i$ indicates the number of the move. So $M_1$ is the first move of the
dialog, $M_2$ the second one, etc. The dialog move $M_i$ is a 5-tuple, where

- $P$ identifies the player. In the examples above Player1 was bert and Player2 was ernie;
- $A$ indicates the illocutionary act, being one of a four element set:
  1. claim: player $P$ claims the formula $S$;
  2. question: player $P$ questions the formula $S$;
  3. accept: player $P$ accepts the formula $S$;
  4. withdraw: player $P$ withdraws the formula $S$;
- $S$ is the propositional content of the illocutionary act. It is a formula of the language
  Language. In the remainder $S$ is often referred to as a sentence, also in case $S$ is a non-
  closed formula.\(^7\)
- $L$ is the level of the move. How the levels change is elaborated in the rules.
- $B$ is the sentence the move $M_i$ provides an argument for, or is a reaction to.\(^8\)

The initial level is zero. Because the level is a positive integer, the term higher level is
ambiguous. I will use the following terminology. The initial level (0) is called the highest
level. The other levels are referred to as follows:

$L < L'$ - $L$ is a higher level than $L'$;
$L' < L$ - $L'$ is a lower or deeper level than $L$.

3.3 The commitment store

Through certain moves players become committed to sentences. Commitment restricts a
player in his subsequent moves, and commitment of the opponent can be used to force him
to accept, or withdraw sentences. Therefore, it is important to store commitments. The
commitment store is defined as follows.

\(^7\) A formula or a term is closed if it does not contain free variables, otherwise the term or formula is
non-closed.

\(^8\) The illocutionary acts question, accept, and withdraw are always reactions. A claimed sentence can
be both a reaction to and an argument for another sentence. When a claim by one player is followed
by a claim of the other player, the second claim is a reaction to the first claim. In all other cases the
claimed sentence is considered to support another sentence, so considered an argument for the
other sentence (See Chapter 6 for the meaning of the word argument in this context).
Definition 3 - the commitment store

A commitment store $C_i (i \geq 0)$ is a set of ordered pairs $(P, S)$, where $P \in \{ \text{Player1}, \text{Player2} \}$, and $S$ a formula of Language.

$C_0 = \emptyset$.

Definition 4 - the set of disputed sentences

The set of disputed sentences $O_i$ is the subset of $C_i$, with the property that

$(P, S) \in O_i$, if and only if

$(P, S) \in C_i$ and $(P', S) \not\in C_i$.

The commitment store exists of the ordered pairs $(P, S)$. A pair means that the player $P$ is committed to the sentence $S$. $C_0$ is defined empty. How the content of the commitment store is settled after each move is elaborated in rule 1.

The set of disputed sentences $O_i$ is the subset of $C_i$ that contains the sentences the players do not agree upon. As we will see later, the dialog continues as long as there are still elements in $O_i$. Since the sentences in $O_i$ are open for discussion, these sentences are also referred to as open sentences.

Note that the set of sentences to which both players are committed is contained in the following set of pairs: $C_i \setminus O_i$. Namely, if the set of disputed sentences is deleted from the commitment store, all pairs with sentences to which only one player is committed are removed. What is left are the pairs with sentences both players are committed to, so a set of undisputed sentences.

3.4 The dialog

Not only the effects of moves in terms of commitment are stored, but also the moves themselves. This record of moves is defined as the dialog.

Definition 5 - the dialog

A dialog $D_i$ is a totally ordered set of $i \geq 1 \leq \text{last}$ elements $M_i$.

$D_1 = (M_1), \text{where } M_1 = (\text{Player1}, \text{claim}, S, 0, \text{dialog}).$

$D_n = D_{n-1} \cup (M_i) \ (1 < i < \text{last}).$

$D_{I_{\text{last}}} = D_{I_{\text{last-1}}} \cup (M_{\text{last}})$, only if $O_{I_{\text{last}}} = \emptyset$.

$D_{I_{\text{last}}}$ is called a finished dialog. $D_i (i < \text{last})$ is called unfinished.

A dialog is defined as a set of moves. $D_1$ consists of the first move, which is defined as follows. Player1 claims a sentence $(S)$, and the level is 0. Since this is the first move, B is not a reaction to, nor an argument for a previous sentence, but conventionally defined as dialog.

Every dialog $D_i (i>1)$ results from adding a move $M_i$ to an already existing dialog $D_{i-1}$. Which moves exactly can be added to a particular dialog is subject of the rules of the game. $D_{I_{\text{last}}}$ is the only dialog to which no moves can be added. $D_{I_{\text{last}}}$ originates if the consequence of a move $(M_{I_{\text{last}}})$ is that the set of disputed sentences $(O_{I_{\text{last}}})$ becomes empty. This occurs when there is agreement about the sentence that was claimed in the first move. Only then a dialog is considered finished.
3.5 The dialog tree

The dialog tree represents the sentences of the game pictorially. The picture provides a good overview of both the history of the game and the commitments. A dialog tree has the following elements:

1. Boxes with either no border, a dotted border, or a solid border:

   no_border  dotted_border  solid_border

2. Sentences (s) inside the boxes;
3. Levels (L) at which the boxes are located. The box at level 0 is the top of the tree. A box at level L+1 is lower in the tree than the box at level L.
4. Arrows pointing from one box to another box.

The meaning of the different borders of the boxes is as follows. If none of the players is committed to a sentence, the box has no border. If only one player is committed, the box has a dotted border. In case both players are committed to a sentence, the box has a solid border.

An example of a tree is the following. It is based on the dialog Bert and Ernie had about O.J. in section 2.6. For the sake of the example the sentences are kept simple.

```
0
   murderer_OJ
  / \      /
1   shot_her  on_purpose_so_murderer
    / \        /
2   smoking_gun
```

The tree is explained as follows. The numbers on the left indicate the levels. If a sentence is claimed, a dotted box with the claimed sentence is added to the picture. So after the first move the dialog tree consisted of only the dotted box at level 0. In general, each time a sentence is claimed a dotted box is added. For instance, after Bert claimed that O.J. shot his wife, the box with shot_her was added. The arrow points to murderer_OJ, because shot_her is an argument for the sentence murderer_OJ. The reason this box has a solid border in the current picture, is because Ernie accepted the sentence shot_her in the fifth move. So, if both players agree about a sentence, the border of the related box becomes solid. The box with smoking_gun has no border. This is because the border of a box vanishes if none of the players is committed. The reason why none of the players is committed to smoking_gun, will become clear in section 4.2.

The tree shows at a single glance the following:

- all the sentences claimed in the dialog;
- at what level the sentences were claimed;
• what sentence an adduced sentence reacts to or means to justify;
• whether both, one or no player(s) is (are) committed to a particular sentence.

The next definition defines the dialog tree.

**Definition 6 - The dialog tree**

\( D_T \) consists of a dotted box on the level 0 with the sentence \( S \) that was claimed in the first move. A dialog tree \( D_T, (i - 1) \) is equal to \( D_T, i \), except in the following cases:

a. if \( M_i = (-, \text{ claim}, S, T, B) \),
   a dotted box with the sentence \( S \) is added, on the level \( T \), with an arrow pointing to the box with the sentence \( B \);

b. if \( M_i = (-, A, S, - , -) \), where \( A \in \{\text{accept, withdraw}\} \),
   the dotted boxes with the sentence \( S \) change in the following cases:
   1. the border vanishes if \( (-, S) \notin C_i \);
   2. the border becomes solid if \( (-, S) \in C_i \cup C_o \).

The dialog tree of the first move contains only a dotted box with the sentence \( S \) that was claimed in the first move. Consecutive trees remain unchanged if the illocutionary act is question. In case a sentence is claimed, a dotted box is added at the level on which it is claimed. An arrow points from the box with the sentence just claimed to the box with the sentence it is a reaction to or argument for (definition 6a). If the act is either accept or withdraw, then some of the dotted borders may change. A border vanishes if the players are no longer committed to the sentence in the box (definition 6b1); a border becomes solid if both players are committed to the sentence in the box (definition 6b2).

Since the rules of the game are not introduced yet, not all features of the tree can be shown here. In the remainder of the chapter the tree is used to exemplify rules.

### 4. General rules for communication

How a dialog actually runs depends on the moves of the players. Recall (see definition 2) that a move is: \((P, A, S, L, B)\). The rules determine which player is to move (\( P \)), and how the level \( L \), and the variable \( S \) change after a move. Within certain constraints the player of a move can choose an illocutionary act (\( A \)) and a propositional content (\( S \)). What exactly is allowed depends on commitment, and on previous moves.

The rules in this section are a basic set. Rule 1 deals with commitment; rules 2-5 are rules for interaction. In section 6 the set of rules is extended (by rules 6-16) with language elements particularly suited for legal discussions. The purpose of the current section is to introduce the basic mechanism of communication between the players. If one is familiar with this mechanism, it will be easier to grasp the more complicated rules related to the special legal language elements.

The rules are numbered 1, 2, ..., 16. The paragraphs of a rule are numbered a, b, c, etc. If a paragraph has more parts, these parts are numbered 1, 2, 3, etc. If such a part has more elements, these are numbered a, b, c, etc. The rules have the following layout.
Rule 17
a. ...
   1 ...
   2 ...
      a ...
      b ...
   3 ...

b ...
c ...

This rule (rule 17) has 3 paragraphs, referred to as rule 17a, rule 17b, and rule 17c. Rule 17a has three parts, referred to as rule 17a1, rule 17a2, and rule 17a3. Rule 17a2 has two elements, referred to as rule 17a2a, and rule 17a2b.

4.1 The interaction in main lines
A dialog starts with the claim of a sentence. All other sentences are related to this sentence. The main goal of the proponent of the first sentence, Player1, will be to justify this first sentence. He can try to achieve this by adducing sentences that justify the first sentence.

The structure of the dialog is layered. The level of the dialog only becomes deeper after the move question. The level remains the same after claim, and in one special case after the move accept (if a player accepts immediately after a sentence is claimed). The level can only become higher after withdraw and accept.

Each sentence on a lower level is by definition related to a sentence on level 0. It is, however, not the case that each sentence of a lower level is related to all higher-level sentences. For instance, in the sample tree of section 3.5, the level-2 sentence smoking-gun is related to the level-1 sentence shot_her, but not to the level-1 sentence on_purpose_so_murderer. All three sentences are related to the 0-level sentence murderer_OJ.

As long as the second player has not denied a sentence, the first player has the initiative. Each sentence he claims is meant to justify sentences that were questioned by his opponent. The opponent can take over the initiative by denying a claimed sentence. Because even the first claimed sentence can be denied, the opponent can take the initiative almost immediately.

The proponent of a sentence that is questioned gets the opportunity to adduce sentences to support the questioned sentence. So the proponent of the sentence can adduce supporting arguments. The opponent of the sentence cannot adduce arguments against the sentence he questioned. The underlying idea is that the opponent should deny the sentence if he wants to adduce arguments against it. If the opponent denies he becomes the proponent of the denied sentence, and as the proponent of the denied sentence he can adduce arguments supporting this denied sentence. These arguments will be arguments against the sentence that he denied.

---

9 In section 6 the model is extended with special language elements. In the extended model the opponent may adduce counterarguments, also if he has not denied the sentence the counterargument is arguing against.
4.2 Origin of commitment

The first rule works out by which moves and how the commitment store \( C_i \) changes. A player becomes committed to each sentence he claims or accepts. A player loses his commitment after withdrawal. In case of question, the commitment store does not change.

**Rule 1**

a. \( C_i = C_{i-1} \cup \{(P, S)\} \)
   - if \( M_i = (P, \text{claim}, S, -, -) \).

b. If \( M_i \in \{(P, \text{withdraw}, S, -, -), (P', \text{accept}, S, -, -)\} \), and
   - \( M_i = (P, \text{claim}, S, -, -), \text{where } h \in \{i-1, i-2, ..., 1\} \), then
     1. \( C_i = C_{i-1} \cup \{(P', S)\} \), if \( M_i = (P', \text{accept}, S, -, -) \), and
     2. \( h = i-1 \).

   2. In all other cases \( U_i \) is the set of pairs to update the commitment store, so that \( U_i = O_{i-1} \cup O_{i-1} \), and
     a. \( C_i = C_{i-1} \cup \{(P, S)\} \), if
        - \( M_i = (P, \text{withdraw}, S, -, -) \);
     b. \( C_i = C_{i-1} \cup C_i \cup \{(P', S)\} \), if
        - \( M_i = (P', \text{accept}, S, -, -) \).

c. \( C_i = C_{i-1} \) in all other cases than those under a, b.

**Rule 1a.** If a player \( P \) claims a sentence \( S \), the pair \( (P, S) \) is added to the commitment store. Note that because the first move is always a claim of a sentence by Player1 (see definition 5), \( C_1 \) will contain the related pair: \((\text{Player1}, S)\).

**Rule 1b.** If a player \( P' \) accepts a sentence \( S \) in the move immediately following after the move in which \( S \) was claimed \( (h = i-1) \), this player \( P' \) becomes committed to \( S \) (rule 1b1). In all other cases a set to update the commitment store is used (rule 1b2a, and rule 1b2b). The example below illustrates how the set updating the commitment works.

**Rule 1c.** After any other move the commitment store remains the same. Using the current set of illocutionary acts this can only be after question.

**Example of rule 1b2**

Consider the following formalization of the dialog until the sixth move, as presented in section 2.6 (see also section 3.5 for a dialog tree of this dialog). For the sake of clarity of the example, the sentences are kept simple.

\[
\begin{align*}
M_1 & = (\text{Bert, claim, murderer_OJ, 0, dialog}) \\
M_2 & = (\text{Ernie, question, murderer_OJ, 0, murderer_OJ}) \\
M_3 & = (\text{Bert, claim, shot_her, 1, murderer_OJ}) \\
M_4 & = (\text{Ernie, question, shot_her, 1, shot_her}) \\
M_5 & = (\text{Bert, claim, smoking_gun, 2, shot_her}) \\
M_6 & = (\text{Ernie, accept, shot_her, 2, smoking_gun})
\end{align*}
\]

\( M_6 \) is of the type \( (P', \text{accept}, S, -, -) \), so \( M_i = M_6 \).

\( M_i \) is the related claim, so \( M_i = M_6 \).

Because \( h \leq 5 \), a set to update the commitment store is needed (rule 1b2).

\( U_6 = O_5 \setminus O_4 \).
DiaLaw - a dialog game for legal justification

\[ O_5 = \{(\text{Bert, smoking\_gun}), (\text{Bert, shot\_her}), (\text{Bert, murderer\_OJ})\} \]
\[ O_4 = \{(\text{Bert, shot\_her}), (\text{Bert, murderer\_OJ})\} \]

Therefore \( O_4 = \{(\text{Bert, smoking\_gun})\} \)

The act of \( O_4 \) was accepted, so according to rule 1b2b the commitment store \( C_5 \) becomes:
\[ C_5 = C_4 \setminus O_4 \cup \{(\text{Ernie, shot\_her})\} \]
\[ = \{(\text{Bert, murderer\_OJ}), (\text{Bert, shot\_her}), (\text{Ernie, shot\_her})\} \]

Note that \( C_5 = O_5 \) because until move 5 Bert claimed three times, and Ernie only questioned.

The only pair that is still in \( O_4 \) is \( \{(\text{Bert, murderer\_OJ})\} \).

In [Lodder & Herzog, 1995] after acceptance or withdrawal open sentences were taken care of differently from the way it is done here. If a player \( P \) accepted a sentence \( S \), then he became committed to all still disputed sentences that were claimed by his opponent after the claim of the sentence \( S \).

Furthermore, the player \( P \) lost his commitment to all still disputed sentences he himself claimed after the claim of the sentence \( S \). In case of withdrawal the open sentences were dealt with in a similar way. The rationale of this rule is that sentences claimed between the claim of a sentence \( S \) and its acceptance or withdrawal, are (considered to be) direct, or indirect arguments for (or against) \( S \). In that perspective agreement about \( S \) was regarded to imply commitment to disputed statements added as arguments for \( S \), and loss of commitment to disputed statements added as arguments against \( S \). However, since irrelevant statements could have been claimed, this forced commitment was not entirely satisfactory.

In the new rule proposed here, the commitment store is updated after acceptance or withdrawal in the following way. The set to update the commitment store contains open sentences - that were claimed after the claim of the sentence \( S \) and before this sentence is accepted/withdrawn. All these sentences are deleted from the commitment store. The reason these sentences are deleted is that they were claimed to justify sentences on higher levels that are no longer subject of the dialog anymore, and probably have become irrelevant. If these sentences would not be deleted, an actually finished dialog (agreement about the 0-level sentence) would not be finished because of open sentences on lower levels.\(^{10}\)

An alternative would be to define a finished dialog as one in which the first-claimed sentence is no longer disputed. In that case a dialog can finish, even if open sentences on lower levels still exist. However, an unpleasant consequence of this approach would be that acceptance/withdrawal of sentences that are no longer relevant would be possible. Deleting the sentences that are no longer relevant is more in spirit of the game. Why store

\(^{10}\) The dialog is defined finished if there are no longer disputed, open sentences. Assume a player claims on level 3 a sentence that convinces his opponent of the main, 0-level sentence. If he in reaction to the sentence claimed on level 3 accepts the 0-level sentence, there is at least one open sentence on level 3, one open sentence on level 2, etc. If these sentences would not be removed from the commitment store, the dialog would not be finished according to the definition of a finished dialog (no open sentences). According to the current commitment rule the level-3, level-2, etc., sentences will be deleted, and as a consequence the dialog will be finished.
commitment to no longer relevant disputed statements, if the purpose of the game is to reach agreement.\footnote{A possible reason could be that this commitment is stored in view of future dialogs. However, in the current version of Dial:law the commitment stores are empty at the beginning of the game, so possible previous dialogs are not taken into account.}

If a player wants to claim a sentence that was deleted, the further rules do not forbid that these sentences are claimed again. This makes it possible that there are more moves in which \( S \) was claimed. In that case the last one of these moves is \( M_n \), reason why \( b \) has to be examined starting with \( i-1 \).

### 4.3 General conditions

In the second rule necessary (but not sufficient) conditions are formulated for all acts, except question. Every move of the game, with the exception of moves with the act question, must obey rule 2.

**Rule 2**

a. \( M_i = \langle P, \text{claim}, S, -, - \rangle \) is only possible, if 
\( (-, S) \notin C_{i-1} \).

b. If \( (-, S) \in C_{i-1} \), then \( M_i = \langle P, \text{claim}, -S, -, B \rangle \) is only possible, if 
\( B = S \).

c. \( M_i = \langle P, \text{accept}, S, -, - \rangle \) is only possible, if 
\( (P', S) \in Q_{i-1} \), and \( (P, -S) \notin C_{i-1} \).

d. \( M_i = \langle P, \text{withdraw}, S, -, - \rangle \) is only possible, if 
\( (P, S) \in C_{i-1} \).

**Rule 2a.** It is not allowed to claim a sentence that is included in the commitment store. This rule looks at three different situations. First, if both players are already committed to a sentence, this sentence may not be claimed again. In that case there is no reason to start a dialog about this sentence. Second, if only the claiming player is committed, the player is not allowed to claim the same sentence again. If this would be allowed, a useless repetition of moves could occur. Third, if only the opponent of the claiming player is committed, the claim is forbidden because in that case acceptance of the sentence would be the appropriate move.

**Rule 2b.** This paragraph of rule 2 forbids contradiction. If a player is committed to a sentence, he is not allowed to claim the opposite. A player is also not allowed to claim a sentence if his opponent is committed to the opposite, except for one situation. In case his claim of \(-S\) is a reaction to the claim of \( S \) (\( B\-S \)). In all other situations such a claim is forbidden, because it would destroy the structure of the argumentation. The following tree demonstrates the strange structure that would result if the latter claim would be allowed.
At level \( n \) it is claimed that Jack is punishable, which is supported with the sentence that Jack is a murderer. The latter sentence is denied. In order to support the sentence that Jack is not a murderer, it is adduced that Jack should not be punished. It is this circle-like argumentation that the rule does not allow. Moreover, if the other player wanted to deny that Jack should be punished, he should have done it in the first place, at level \( n \).

**Rule 2c.** A player can only accept a sentence if he is not committed to that sentence, and the other player is. This can only be the case if the other player claimed this sentence. Moreover, it is not allowed to accept a sentence if the player is committed to the negation of that sentence. If a player would be allowed to do this, the commitment store would contain \( S \) and \( \neg S \) for the same player; direct contradiction is forbidden in the game (see section 1).

**Rule 2d.** A player can only withdraw a sentence if he is committed to that sentence, and the other player is not. This means that he previously claimed this sentence, and the sentence is not yet accepted.

### 4.4 Moves after a claim

The third rule deals with the moves that can follow after a claim.

**Rule 3**

\[
\text{If } M_{n-1} = (F, \text{ claim }, S, L, -) \text{ then } \\
M_n = (F', A, T, L, S), \text{ where} \\
A \text{ can be any act, and} \\
T \text{ can be any sentence, except that} \\
T = S, \text{ if } A = \text{ question, and} \\
T = \neg S, \text{ if } A = \text{ claim.}
\]

The level remains the same after a claim, and the next move is a reaction to the sentence \( S \). Basically, the player \( p' \) can react to a claim with any act. Besides the general conditions of rule 2 that apply to the move by player \( p' \), this rule defines two specific constraints. First, if the act is question, the propositional content is the sentence claimed in the previous move, \( S \).

Second, a claim as reaction to a claim is only allowed if the claimed sentence (\( \neg S \)) is the negation of the previous claimed sentence (\( S \)). If just any claimed sentence was allowed, two problems would occur. First, the dialog could consist of an infinite sequence of claims. Second, and most importantly, the relation between the sentences would disappear. In the current model each claimed sentence supports a sentence claimed on a higher level (or at
least is assumed to do so) or denies a previously claimed sentence. Each move is a reaction to another move. Whether sentences have to be justified depends on the reaction of the other player; DiaLaw is a reaction-based game. A claim can only be guaranteed to be a reaction if there is a link with a previous move. In the current model this relation is guaranteed: a claimed sentence is either a reaction to a demand for justification (question) or a reaction to a claimed sentence (denial).

4.5 Moves after question

The fourth rule is about the only situation in which the level of the dialog becomes lower when a sentence is questioned. The next move is always by the other player. This next move is an argument for the questioned sentence \( \varepsilon \), in case the next move is a claim. Accept and withdraw are, as always, also allowed. In that case the move is a reaction to \( \varepsilon \). The player \( P' \) may not question, for this would mean that he questions a question.

**Rule 4**

If \( M_{i-1} = (P, \text{question}, S, L, \_\_\_) \), then

\[ M_i = (P', \text{A}, \_\_, L_i +1, S) \], where A ≠ question.

4.6 Moves after acceptance or withdrawal

The fifth rule is about the move following accept or withdraw. It is the only move after which the level of the dialog can become higher. Moreover, it is only after accept and withdraw that sometimes the same player moves again. Accept and withdraw are taken together, because their successors are largely the same. If a player accepts or withdraws a sentence, the dialog returns to the level the withdrawn/accepted sentence was claimed at. Basically, the next move is either a new argument for, or a new reaction to the same sentence as the accepted/withdrawn sentence was an argument for or a reaction to.

**Rule 5**

a. If \((P', -S) \in C_{i-1}\), and

\[ M_{i-1} = (P, \text{withdraw}, S, \_\_, \_\_) \], then

1. \[ M_i = (P, \text{question}, -S, L, \_\_\_\_) \] or
2. \[ M_i = (P, \text{A}, \_\_, L, -S) \], where A ∈ (accept, withdraw).

The level \( L_i \) in \( M_i \) is the same as in the element \((P, \text{claim}, S, L, \_\_) \) \( \in D_{i-1} \).

b. If \((P', -S) \notin C_{i-1}\), and

\[ M_{i-1} \in \{(P', \text{accept}, S, \_\_, \_\_), (P, \text{withdraw}, S, \_\_, \_\_\_)\} \], then

by using \( B \) from the element \((P, \text{claim}, S, L, B) \) \( \in D_{i-1} \), the following two cases can be distinguished:

1. If A ≠ -S, then

\[ M_i = (P, \text{A}, \_\_, L, B) \], where A ≠ question;

2. Otherwise (so, B = -S),

\[ M_i = (P', \text{A}, \_\_, L, B') \], where A ≠ question, and B' is the same as in element

\((P', \text{claim}, -S, L, B') \) \( \in D_{i-1} \).

Note that the level \( L \) in \( M_i \) is the same as in the element \((P, \text{claim}, S, L, B) \).
DialLaw - a dialog game for legal justification

This rule distinguishes in the first place between two situations: when the negation (\(-S\)) of the accepted/withdrawn sentence \(S\) is in the commitment store (rule 5a), and when it is not (rule 5b).

**Rule 5a.** This paragraph of rule 5 is only about withdraw, not about accept. The following trees represent the situation in the dialog, as far is relevant to rule 5a, at the moment when move \(M_i\) is about to be made. The sentence \(S\) is just withdrawn.

\[
\begin{array}{c}
 S \\
 S \\
 \hline
 -S \\
 -S \\
\end{array}
\]

If a player \(P\) withdraws a sentence \(S\) and his opponent is committed to the negation of that sentence \((-S)\), this player \(P\) must react in the next move to \(\neg S\). The rule looks at two different situations. The following example shows one of them (see also the left tree).

- \((P, \text{claim, murderer(oj)}), 2, -\)
- \((P', \text{claim, \neg murderer(oj)}), 2, \text{punish(oj)}\)
- \((\ldots)\)
- \((P, \text{withdraw, murderer(oj)}, 2+m, -), \text{where } m \geq 1\)
- \((P, A, -, 2, \neg\text{murderer(oj)})\)

The player \(P\) claimed that O.J. is a murderer \((S)\) and the other player \(P'\) denied this \((-S)\). Later in the game the player \(P\) withdraws that O.J. is a murderer \((S)\). In the next move, which is a reaction to the claim of \(\neg\text{murderer(oj)}\), he is allowed to question the sentence \(\neg\text{murderer(oj)}\). He may also withdraw another sentence or accept a sentence (including the sentence \(\neg\text{murderer(oj)}\)). Since the only allowed counter claim is a denial, and he already claimed the denial of this sentence (that he just withdrew), he is not allowed to claim.

The other situation (see the right tree) is that in which at the last but one move the player \(P'\) withdraws \(\neg\text{murderer(oj)}\). The next move would then be a reaction to \(\text{murderer(oj)}\):

- \((P', A, -, 2, \text{murderer(oj)})\).

**Rule 5b.** The second paragraph of rule 5 deals with the situation in which a player withdraws or accepts a sentence, while the other player is not committed to the negation of that sentence. First rule 5b1 and then rule 5b2 is discussed. It will be explained whose turn it is, and what the next move will be about \((S)\).

The following trees represent the situation in the dialog, as far is relevant for rule 5b1, at the moment move \(M_i\) is about to be made.

---

\[12\] The related move for accept is: \(M_{i+1} = (P', \text{accept, } S, -, -)\). Since the pair \((P', -S)\) is contained in the commitment store (one of the conditions of rule 5a), this accept move is forbidden by rule 2c. That is why this (impossible) move is not dealt with in rule 5a.
In each of the five cases it is the player who previously claimed \( B \) who continues. In the move that follows \( (M_i) \) he may claim (a new sentence supporting \( B \)), accept or withdraw. He is only not allowed to question.

In case the just withdrawn sentence \( S \) was not claimed as a reaction to the claim of \( \neg S \) (rule 5b2), the withdrawing player continues. The most left tree shows the dialog in which the withdrawn sentence \( S \) was not previously denied; the most right tree shows the dialog in which the sentence was denied \( \neg S \) and the denial subsequently withdrawn.

In case the sentence \( S \) is accepted, the opponent of the accepting player continues. The two trees in the middle show the dialogs in which the accepted sentence was not previously denied (second tree) and the dialog in which it was (third tree).

The following trees represent the situation in the dialog, as far is relevant for rule 5b2, at the moment the move \( M_i \) is about to be made.

It is always the player who claimed the sentence \( B' \) that continues, which is the same player as the one who claimed \( \neg S \), in the move \( M_i \) he may claim (a new sentence supporting \( B' \)), accept or withdraw. He is only not allowed to question.

The left tree shows the dialog in which \( S \) was just withdrawn. The opponent of the withdrawing player continues. The right tree shows the dialog in which \( S \) was just accepted. In case of acceptance the accepting player moves again. The following example, again starting at level 2, is example of the left tree.

\[
\begin{align*}
(P, \text{claim, murderer}(o_j), 2, \text{punish}(o_j)) \\
(P', \text{claim, } \neg\text{murderer}(o_j), 2, \text{murderer}(o_j)) \\
(\ldots) \\
(P, \text{withdraw, murderer}(o_j), 2+m, \_), (m \geq 1) \\
(\ldots) \\
(P', \text{withdraw, } \neg\text{murderer}(o_j), 2+m, \_), (n \geq 1) \\
(P, A, \_ , 2, \text{punish}(o_j))
\end{align*}
\]

The player \( P \) claimed \( \text{murderer}(o_j) \) as an argument for \( \text{punish}(o_j) \), and the other player \( P' \) denied this sentence. Later in the game, the player \( P \) withdraws his sentence. Again later the player \( P' \), who previously denied, also withdraws his sentence. The dialog continues with a move by the other player \( P \). In this move he may claim (a new sentence supporting \( \text{punish}(o_j) \)), accept or withdraw. He is only not allowed to question. The
following tree represents, as far as can be known, the situation of the example before the player p makes the last move.

```
1  punish(oj)
   ^
2  murderer(oj)  ^
      - murderer(oj)
```

The last move would have been the same if in the last but one move \(~\text{murderer}(oj)\) was accepted, so if the last but one move would have been:

\((P, \text{accept}, \text{~murderer}(oj), 2+n, \text{~}-(n \geq 0))\).

In the above tree the box with \(~\text{murderer}(oj)\) would then have a solid border (see also the right tree of rule 5b2).

5. **SPECIAL LANGUAGE ELEMENTS**

In this section special elements of the language are introduced. The elements are adaptations of the elements described in [Lodder & Herzog, 1995] and stem, with one exception, from Reason-Based Logic [Hage & Verheij, 1994]. In the next section special rules for communication will be defined using these special language elements.

The special language elements are the function symbol, \text{rule}/2, and five predicate symbols: \text{reason}/2, \text{outweighs}/3, \text{excluded}/1, \text{applies}/1, and \text{valid}/1. Before discussing the different symbols, I will briefly introduce them informally. In addition to these Reason-Based Logic related elements, there is also a particular dialogical predicate symbol: \text{il_claim}/1.

Before explaining all these elements, first a brief introduction is given of Reason-Based Logic as it is used in DiaLaw."13

5.1 **Introducing DiaLaw’s Reason-Based Logic**

A reason (\text{reason}/2) is a support relation between the states of affairs expressed by sentences. If the sentence \text{murderer}(oj) supports the sentence \text{punish}(oj), then \text{murderer}(oj) can be presented as a reason for \text{punish}(oj). A sentence is justified if the reasons for the conclusion outweigh (\text{outweighs}/3) the reasons against the conclusion. In case there are only reasons pro, this weighing is trivial: the sentence supported by the reason(s) is justified. In case reasons point in opposite directions justification of a sentence depends on which set of reasons is preferred. If the set of reasons for a sentence is preferred above the set of reasons against the sentence, the sentence is justified. If the preference is the other way around, the sentence is not justified.

---

13 For further reading on Reason-based Logic consult [Hage 1996, 1997a] and [Verheij, 1996].
Reasons can be supported by underlying rules or principles, both referred to as rule
(rule/2). Although not necessarily each reason has an underlying rule, the application
(applies/1) of a rule leads to the related reason. Suppose there is a rule saying that
someone who murders must be punished. If this rule applies, the related reason is justified.
For instance, if the above rule applies for O.J., then the following reason is justified.

\[
\text{punish(oj)} \\
\text{outweighs(Reason_set_pro, Reason_set_con, punish(oj))}
\]

When does a rule apply? Basically, if the reasons for applying the rule outweigh the reasons
against applying it. This weighing will often be trivial, because in many cases there will be
only one reason for applying the rule: if a rule is valid (valid/1) and the condition of the rule
is satisfied, there is a reason to apply it. However, there can also be reasons not to apply the
rule, for example, that application would be contrary to the purpose of the rule.

\[
\text{applies(rule(murderer(oj), punish(oj)))} \\
\text{outweighs(Reason_set_pro, Reason_set_con, applies(rule(murderer(oj), punish(oj))))} \\
\text{by the constitue one reason applies} \\
\text{valid(rule(murderer(X), punish(X))), and} \\
\text{murderer(oj)}
\]

There can be reasons for and against applying a rule that need to be weighed in order to
decide whether a rule applies. This weighing becomes irrelevant if a rule is excluded
(excluded/1). Then the rule simply does not apply. In case a rule is excluded, the
conclusion that the rule applies is no longer justified. The rule about murder for example is
excluded in case the murder was committed in self-defense.

\[
\text{excluded(rule(murderer(oj), punish(oj)))} \\
\text{makes justification impossible} \\
\text{applies(rule(murderer(oj), punish(oj)))}
\]

\(^4\) In Reason-Based Logic each reason has an underlying rule. In [Lodder, 1997a] it is argued that
reasons do not necessarily have underlying rules.
5.2 Formal properties of the special language elements

Below the formal properties of all special language elements are discussed.

- \textit{il\_claim(S)}

  The predicate \textit{il\_claim} says that some sentence \textit{S} was claimed illegally. The formula \textit{il\_claim} is used when a player thinks that a particular sentence may not be claimed. This is different from the case in which the dialog rules forbid a particular claim. In the case of \textit{il\_claim} the rules do not forbid the claim of this particular sentence, but there are other-domain dependent-reasons why the sentence may not be claimed. For instance, if a player has claimed a sentence based on illegally obtained evidence. In that case justification of the sentence is irrelevant, because in law sentences based on false evidence may not be used.\footnote{If the judge is convinced that evidence is illegally obtained, then he will not take into account the information that was obtained by the evidence.}

- \textit{reason(Cond, Concl)}

  This formula says that \textit{Cond} is a reason for \textit{Concl}, where \textit{Cond} and \textit{Concl} are both closed terms. From a different perspective, the formula can also be seen as expressing that \textit{Cond} is a reason against \neg \textit{Concl}. The following formulas are examples of reasons:

  \begin{center}
  \textit{reason(murderer(oj), punish(oj)),}
  \textit{reason(falsely\_accused(oj), \neg\textit{punish(oj)})}.
  \end{center}

  The fact that O.J. is a murderer is a reason to punish him (or a reason against not punishing him). The fact that O.J. is falsely accused is a reason not to punish him (or a reason against punishing him).

- \textit{outweighs\{\textit{Cond}\textsubscript{pro\_1}, \textit{Cond}\textsubscript{pro\_2}, ..., \textit{Cond}\textsubscript{pro\_m},
  \textit{Cond}\textsubscript{con\_1}, \textit{Cond}\textsubscript{con\_2}, ..., \textit{Cond}\textsubscript{con\_n},
  \textit{Concl}\} (m\geq1, n\geq0)}

  The formula \textit{outweighs} should be read as follows. The set of closed terms \{\textit{Cond}\textsubscript{pro\_1}, ... , \textit{Cond}\textsubscript{pro\_m}\} originates from the sentences \textit{reason(Cond}\textsubscript{pro\_1}, \textit{Concl}); the set of closed terms \{\textit{Cond}\textsubscript{con\_1}, ... , \textit{Cond}\textsubscript{con\_n}\} originates from the sentences \textit{reason(Cond}\textsubscript{con\_1}, \neg \textit{Concl}). The set of reasons pro must contain at least one term (m\geq1), the set of reasons con may be empty (n\geq0). For example, assume both players are committed to the following reasons:

  \begin{center}
  \textit{reason(thief, punish),}
  \textit{reason(payed\_damage\_victim, \neg\textit{punish}), and}
  \textit{reason(has\_job, \neg\textit{punish}).}
  \end{center}

  In that case, depending on which set is preferred, the following two formulas can be claimed:

  \begin{center}
  \textit{outweighs\{(thief),}
  \{payed\_damage\_victim, has\_job\},
  \textit{punish\}}
  \textit{outweighs\{payed\_damage\_victim, has\_job\},
  \{thief\},
  \neg\textit{punish\}}
  \end{center}
• rule(Cond, Concl)
The function rule means that some rule has a condition Cond and a conclusion Concl. Both Cond, Concl are terms that may contain free variables.

• valid(rule(Cond, Concl))
The predicate valid says that some rule is valid. Cond and Concl are terms that may contain free variables.

• applies(rule(Cond, Concl))
The predicate applies says that some rule applies. Cond and Concl are closed terms. If a rule applies, the reason based on that rule is justified.

• excluded(rule(Cond, Concl))
The predicate excluded says that some rule is excluded. Cond and Concl are closed terms. If a rule is excluded, it cannot apply.

6. Special rules for communication - legal tools and forced commitment

The following special rules for communication (rules 6-16) extend the general rules for communication (rules 1-5). The special rules formulate constraints for the players, related to the special language elements introduced in the previous section. Moreover, an extra feature is added to the game: forced commitment. Justification in DiaLaw is based on agreement. If a player claims that murder is legal, and the other player accepts this sentence, the sentence is justified. Acceptance is normally based on free choice. However, if special elements of the language are used, players can be forced to accept or withdraw sentences. In other words, players can be forced to become committed or to lose commitment.

The sixth\textsuperscript{16} rule forbids the players to claim sentences they previously withdrew. This constraint is necessary, because it would not be appropriate to allow a player to claim a sentence, if he was forced to withdraw it earlier. In order to complicate the matter not too much, a player who voluntarily withdrew a sentence is not allowed to reclaim it either. So each player must be aware of this consequence wherever he withdraws a sentence.

Rule 6

\[ m = (P, \text{claim}, S, -, -) \text{ is only possible, if} \]
\[ (P, \text{withdraw}, S, -, -) \notin \Sigma_i. \]

6.1 On il_claim

This section is about the special predicate il_claim. Basically, a claim as a reaction to the claim of a sentence S is only allowed if this sentence is denied, so if \text{~S} is claimed. One of the exceptions to this general rule is that a player may react to a claim of a sentence with the claim that it is illegal to claim that particular sentence. The seventh rule is about this

\textsuperscript{16} Rules 1-5 are discussed in section 4.
situation. The eighth rule regulates consequences of accepting and withdrawing the sentence il_claim.

**Rule 7**

\[ M_i = (P', \text{claim}, \text{il_claim}(S), L, S) \text{, only if} \]

\[ M_i \rightarrow = (P, \text{claim}, S, L, B) \text{, where} \]

\[ S \neq \text{il_claim}(\cdot) \]

This rule formulates conditions for the claim of the sentence il_claim(\cdot). First, the claim must follow a claim by the opponent. Second, the sentence of this other claim may not be il_claim(\cdot). The reason why this is forbidden, is because otherwise the claim of the meaningless sentence il_claim(il_claim(\cdot)) would be possible.

**Rule 8**

Under a, b, and c, use the element \((P, \text{claim}, S, L, B) \in D_{0,1}\).

a. If \((P', \text{il_claim}(S)) \in O_{i-1}\), and

\[ M_i \rightarrow = (P, \text{withdraw}, S, \_, \_), (P', \text{accept}, S, \_, \_), \text{then} \]

\[ M_i = (P', \text{withdraw}, \text{il_claim}(S), L, S) \text{, and} \]

\[ M_{i-1} = (P, A, \_, L, B) \]

For \(M_i\), the following constraints hold:

1. If \(B = \neg S\), then
   a. \(A \neq \text{claim}\,\text{and}\)
   b. \(A \neq \text{question}, M_{i-1} = (P, \text{question}, \neg S, L, \neg S)\)

2. Otherwise (so \(B \neq \neg S\)) \(A \neq \text{question}\).

b. If \((P', \text{il_claim}(S)) \in O_{i-1}\), and

\[ M_i \rightarrow = (P, \text{accept}, \text{il_claim}(S), \_, \_), \text{then} \]

\[ M_i = (P, \text{withdraw}, S, L, \text{il_claim}(S)) \]

c. If either,

\[ M_{i-1} = (P', \text{withdraw}, \neg \text{il_claim}(S), \_, \_), \text{and} \]

\[ (P', \text{il_claim}(S)) \notin O_{i-1} \text{, or} \]

\[ M_{i-1} \in \{(P, \text{withdraw}, \neg \text{il_claim}(S), \_, \_), (P', \text{accept}, \neg \text{il_claim}(S), \_, \_), \text{and} \}

\[ (P', \text{il_claim}(S)) \notin O_{i-1} \text{, then} \]

1. \(M_i = (P', \text{question}, S, L, S)\)
2. \(M_i = (P', A, \_, L, S)\,\text{where} \]

\[ A \in \{\text{accept, withdraw}\} \]

**Rule 8a.** The first paragraph of rule 8 is about the situation in which a discussion about il_claim(S) is still open, and the sentence S is either accepted or withdrawn. No matter whether player P withdrew S or player P' accepted this sentence, in the next move the player P' is forced to withdraw il_claim(S). The reason why he is forced, is that a discussion about this sentence is no longer relevant. Namely, if the player P has withdrawn S, the effect desired by the claim of il_claim(S) is already obtained; if player P' accepted S, his new commitment to S cannot be combined with il_claim(S). After the
forced withdrawal, the discussion can continue in two different ways. First, if the sentence S was claimed to deny ¬S,

\[ ¬S \rightarrow S \leftarrow \text{il_claim}(S) \]

the discussion continues as indicated in rule 5a: the player P may question ¬S, accept or withdraw. He only is not allowed to claim (rule 8a1).17

Second, in case the sentence S was not a denial of the sentence ¬S,

\[ \begin{array}{c}
\text{n} \\
\uparrow \\
\text{n+1} \\
\uparrow \\
\text{s} \leftarrow \text{il_claim}(S)
\end{array} \quad \begin{array}{c}
\text{n} \\
\uparrow \\
\text{n+1} \\
\uparrow \\
\text{B} \leftarrow \text{il_claim}(S)
\end{array} \]

the dialog continues as indicated in rule 5b: the player P may claim (a new sentence supporting B), accept or withdraw. He only is not allowed to question (rule 8a2).

**Rule 8b.** The following situation is dealt with.

\[ \begin{array}{c}
\text{n} \\
\uparrow \\
\text{n+1} \\
\uparrow \\
\sim \leftarrow \text{il_claim}(S)
\end{array} \]

If the sentence \( \text{il_claim}(S) \) is accepted by the player P, he is forced to withdraw S. The withdrawal is forced, because commitment to both the sentences S and \( \text{il_claim}(S) \) is incompatible.

**Rule 8c.** This paragraph of rule 8 deals with the move following the withdrawal of \( \text{il_claim}(S) \), or the withdrawal/acceptance of \( ¬\text{il_claim}(S) \).

---

17 The sentence ¬S is still open. In case the sentence ¬S is not open because it was withdrawn before S was either accepted or withdrawn, the discussion would continue with a discussion about S (see rule 5a). A consequence of this withdrawal of ¬S would be that the discussion about \( \text{il_claim}(S) \) ends, since the open sentence would, following rule 1b2, be deleted.
DiaLaw - a dialog game for legal justification

a. \[ S \Rightarrow \text{il\_claim}(S) \]
b. \[ S \Rightarrow \text{il\_claim}(S) \Rightarrow \neg \text{il\_claim}(S) \]
c. \[ S \Rightarrow \text{il\_claim}(S) \Rightarrow \neg \text{il\_claim}(S) \]

One of the first two trees (a, b) originates after \( \text{il\_claim}(S) \) is withdrawn, if \( \text{il\_claim}(S) \) has not been denied, the resulting tree is a. This will normally be the case. If a player claims that a sentence is illegally claimed, it is more probable that he asks for a justification, than that he denies it. However, it is not forbidden to deny. The reason to deny could be to hold the initiative in the dialog. If \( \text{il\_claim}(S) \) was denied, the resulting tree is tree b.

The last two trees (b, c) originate either after \( \neg \text{il\_claim}(S) \) is withdrawn, or after \( \neg \text{il\_claim}(S) \) is accepted. If \( \neg \text{il\_claim}(S) \) is withdrawn, the resulting tree is b. If \( \neg \text{il\_claim}(S) \) is accepted, the resulting tree is c.

The dialog continues after each of the three moves of rule 8c in the same way as indicated in rule 5a: the player \( P \) may question \( S \), accept or withdraw. He only is not allowed to claim.

6.2 On reasons to apply a rule

Whenever a player is committed to the sentence that a particular rule is valid, and he is committed to a sentence that satisfies the condition of this rule, he must accept that there is a reason to apply the rule. The applicability of a rule this reason is called. So if a rule is applicable, which means that it is valid and its condition is satisfied, there is a reason to apply the rule.

Rule 9

Let \( X = \text{Cond} \), where \( \text{Cond} \) is a closed formula,

\[ Y = \text{valid}(\text{rule}(\text{Cond}, \text{Concl})), \]

where \( \text{Cond} \) and \( \text{Concl} \) may contain free variables,

\[ Z = \text{reason}(\text{applicable}(\text{rule}(\text{Cond}, \text{Concl}))), \]

applies(\text{rule}(\text{Cond}, \text{Concl})),

where \( \text{Cond} \) and \( \text{Concl} \) are closed terms, then:

a. \[ M_i = (P^i, \text{accept}, Z, \_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_) \]
   if \((P^i, Z) \in O_{\_\_\_\_}\)
   and
   \((P, X), (P, Y) \in C_{\_\_\_\_}\),

b. If either \((P^i, \_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_) \in C_{\_\_\_\_}\)
   or \((P^i, \_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_) \in C_{\_\_\_\_}\), then
   1. \[ M_i = (P^i, \text{withdraw}, Z, \_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_) \]
   if \((P^i, Z) \in O_{\_\_\_\_}\)
   and
   2. \[ M_i = (P^i, \text{claim}, Z, \_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_) \]
   is not allowed.

Since the condition of the rule \((X = \text{Cond})\) is satisfied, \( \text{Cond} \) must be a closed formula. The predicate \text{valid} means that a rule is valid in general, reason why \( \text{Cond} \) and \( \text{Concl} \) may
contain free variables (see section 5). In case of applicable and applies the rules are initiated, reason why Cond and Concl are closed terms.

**Rule 9a.** The player $P$ is forced to accept the reason that the rule is applicable, if he is committed to that the rule is valid as well as to that the condition of the rule is satisfied.

**Rule 9b.** The second paragraph of rule 9 is about the situation in which a player is committed to either that the rule is not valid or to that the condition of the rule is not satisfied. In case he claimed the reason that the rule is applicable, and this reason is not accepted yet nor withdrawn, he is forced to withdraw this reason. Moreover, it is forbidden for this player to claim the reason that the rule is applicable.

### 6.3 On applies, reason, and excluded

This rule is about the relation between three sentences. First, if a player is committed to the fact that a rule applies, he is forced to accept the reason based on that rule. Second, if a player is committed to the sentence that a rule is excluded, he may not be, nor become committed to the sentence that the rule applies.

**Rule 10**

Let $X = \text{excluded} (\text{rule} (\text{Cond}, \text{Concl}))$,

$Y = \text{applies} (\text{rule} (\text{Cond}, \text{Concl}))$,

$Z = \text{reason} (\text{Cond}, \text{Concl})$, where $\text{Cond}$ and $\text{Concl}$ are closed terms, then:

a. $M_i = (P, \text{accept}, Z, -, -)$, if

$(P', Z) \in C_{i-1}$, and

$(P, Y) \in C_{i-1}$.

b. If $(P, X) \in C_{i-1}$, then

1. $M_i = (P, \text{withdraw}, Y, -, -)$, if $(P, Y) \in C_{i-1}$, and

2. $M_i = (P, \text{claim}, Y, -, -)$ is not allowed.

$\text{Cond}$ and $\text{Concl}$ are closed terms, because all predicates are about initiated rules.

**Rule 10a.** The first paragraph of rule 10 is about a player $P$, who is committed to that a rule applies, and not committed to the reason based on that rule. In case his opponent has claimed the reason based on that rule, the player $P$ is forced to accept it.

**Rule 10b.** The second paragraph of rule 10 is about a player $P$, who is committed to that a rule is excluded. He is forced to withdraw the sentence that the rule applies if he previously claimed this still open sentence. Furthermore, this player is not allowed to claim that the rule applies.

### 6.4 On outweighs and reason

In this section the rules about reasons and the weighing of reasons are formulated. The first rule is about the level at which these sentences may be claimed. Both reason and outweighs are direct arguments for a particular sentence $S$, and therefore have to be claimed on the level that originates after this sentence $S$ is questioned. So, a claim of reason or outweighs about a sentence $S$ is only possible one level deeper than $S$ was claimed (this is only possible if $S$ was questioned and is still disputed).
Rule 11

\[ M_s = \{-, \text{claim}, X, L, S\}, \text{where} \]
\[ X \in \{\text{reason}(-, S), \text{reason}(-, -S), \]
\[ \text{outweighs}(-, -, S), \text{outweighs}(-, -, -S)\}, \text{only if} \]
\[ (-, \text{claim}, S, L-1, -) \in D_{s-1}. \]

Note that because \text{reason} and \text{outweighs} are claimed on the same level, and are arguments for the same sentence, it is impossible that a reason is still open at the moment the outweigh claim is made.

The next rule is about who may claim \text{outweighs}, and in how case of such a claim the sets of reasons are filled. Only the player who is committed to the sentence \( S \) may claim that the reasons related to \( S \) dip the balance in favor of \( S \). The set of reasons for the sentence \( S \), the \text{Proset}, contains all reasons pro that the players agreed upon. The set of reasons against the sentence \( S \), the \text{Conset}, is filled in the same way. Recall that the set of sentences on which the players agree is obtained by deleting the set of open sentences from the commitment store. Because the claim of \text{outweighs} makes only sense if there is at least one reason, the set of reasons pro may not be empty.

Rule 12

\[ M_s = \{P, \text{claim, outweighs(Proset, Conset, S),} -, -\}, \text{only if:} \]
\[ \bullet \ (P, S) \in O_{s-1}; \]
\[ \bullet \ \text{Proset is a set of elements Pro, \{Pro_1, Pro_2, \ldots, Pro_n\}, where} \ n \geq 1. \text{For each} (-, \]
\[ \text{reason(Pro, S)} \in C_{s-1} \setminus O_{s-1}, \text{the element Pro is added to Proset;} \]
\[ \bullet \ \text{Conset is a set of elements Con, \{Con_1, Con_2, \ldots, Con_m\}, where} \ m \geq 0. \text{For each} (-, \]
\[ \text{reason(Con, -S)} \in C_{s-1} \setminus O_{s-1}, \text{the element Con is added to Conset.} \]

The following tree shows the minimal circumstances under which the claim of \text{outweighs} is allowed.

```
\[ n \]
\[ n+1 \]
\[ \text{reason(Pro, S)} \]
\[ \text{outweighs((Pro), ()), S) \]
```

Only reasons that are agreed upon are included in the reason sets of \text{outweighs}. This is in the first place because otherwise the meaning of questioning \text{outweighs} would not be clear. Namely, the question could concern particular elements that are part of the weighing, but also the weighing itself. Furthermore, weighing of a particular set of reasons is rather senseless if there is no agreement about the reasons involved. What is the use of claiming that some set of reasons outweighs another set of reasons, if the other player has not accepted the reasons in the first place?

There is still another reason for not allowing reasons in \text{outweighs} that players did not agree upon. It has to do with the special status of \text{outweighs} without reasons con. Since an empty set has no weight, it is not allowed to deny that the reasons pro outweigh the reasons
against if there are only reasons pro. For the same reason this sentence may not be questioned. This is laid down in the next rule.

**Rule 13**
The following moves are allowed only if \( \text{Conset} \neq \{ \} \):

\[
M_t = (P, \text{question, outweights} \{ \text{Propset, Conset, } S \}, \neg, \neg), \text{and}
\]
\[
M_s = (P, \text{claim, \neg\text{outweights} \{ Propset, Conset, } S \}, \neg, \neg).
\]

The next rule deals with forced acceptance and withdrawal in case outweights is accepted.

**Rule 14**
If \( M_{\text{in}} = (P, \text{accept, outweights} \{ \text{Propset, Conset, } S \}, \neg, \neg) \), then:

\[
M_t = (P, \text{accept, } S, L, S) \text{, if } (P, -S) \in O_{\text{on}},
\]

otherwise (so \( P, -S \in O_{\text{off}} \))

\[
M_t = (P, \text{withdraw, } -S, L, S) \text{, and}
\]

\[
M_{\text{out}} = (P, \text{accept, } S, L, -S).
\]

The level \( L \) is the same as in the element,

\[
(P', \text{claim, outweights} \{ \text{Propset, Conset, } S \}, L, S) \in D_{\text{in}}.
\]

Note that the move in which outweights was claimed is used to determine the levels of the move \( M_t \).

If a player \( P \) becomes committed to an outweights sentence about \( S \), he is forced to accept the sentence \( S \). Moreover, if the player himself is committed to \( \neg S \), he must withdraw this sentence before accepting \( S \). For example, player \( P \) claimed \( \neg S \) (and is still committed), and accepts outweights(\( \neg, \neg, S \)). Then this player must withdraw \( \neg S \) and subsequently accept \( S \).

The last two rules are about reacting to a claim with a claim. In case a player claims that the reasons pro outweigh the reasons con, his opponent is allowed to react to this claim with a reason con.

**Rule 15**
If \( M_{\text{in}} = (P, \text{claim, outweights} \{ \neg, \neg, S \}, L, \neg) \), then

\[
M_t = (P', \text{claim, reason} \{ \neg, S \}, L, \text{outweights} \{ \neg, \neg, S \}) \text{ is allowed.}
\]

The following rule elaborates on the consequences of the claim of a reason as described in rule 15. The basic rules for accept and withdraw do not work here, because their starting point is that only in case of a denial a claim can follow a claim. Rule 16 is comparable to rule 8, because that rule also dealt with consequences of two consecutive claims other than denial.
Rule 16
Let \( \langle P', \text{claim}, X, L, S \rangle \) \( \in D_{k-1}, \) where \( X = \text{outweighs}(-, \neg, S), \) and \( Y = \text{reason}(-, \neg, S). \)

a. If \( M_{k-1} = \langle P, \text{withdraw}, X, \neg, \neg \rangle, \) and \( \langle P', Y \rangle \in O_{k-1}, \) then
   1. \( M_k = \langle P, \text{question}, Y, L, Y \rangle, \) or
   2. \( M_k = \langle P, \text{accept}, \neg, L, Y \rangle, \) where \( A \in \{ \text{accept, withdraw} \}; \)

b. If \( (-, X) \in C_{k-1}, \) and
   \( M_{k-1} = \langle P, \text{accept}, Y, \neg, \neg \rangle, \) then
   \( M_k = \langle P, \text{accept}, \neg, L, S \rangle, \) where \( A \notin \text{question}; \)

c. If \( \langle P, X \rangle \in O_{k-1}, \) and
   \( (-, Y) \notin O_{k-1}, \) then:
   1. \( M_k = \langle P, \text{withdraw}, Y, L, Y \rangle, \) if \( \langle P, Y \rangle \in C_{k-1} \}
   2. otherwise (so \( \langle P, Y \rangle \notin C_{k-1} \}),
      \( M_k \notin \{ \langle P', \text{question}, Y, L, X \rangle, \)
      \( \langle P', \text{claim}, \text{reason}(\neg, \neg), L, X \rangle, \)
      \( \langle P', \text{accept}, \neg, L, X \rangle \}, \) where \( A \in \{ \text{accept, withdraw} \}; \)

d. If \( M_{k-1} = \langle P', \text{accept}, X, \neg, \neg \rangle, \) and \( \langle P', Y \rangle \in O_{k-1}, \) then
   \( M_k = \langle P', \text{withdraw}, Y, L, X \rangle; \) and
   \( M_{k-1} = \langle P', \text{accept}, S, L, S \rangle. \)

Before discussing each paragraph of rule 16, the related tree with only the relative sentences is shown.

Rule 16a.

\[
\begin{array}{c}
X \\
\end{array} \quad \begin{array}{c}
Y \\
\end{array}
\]

In case \text{outweighs} is withdrawn \( X \), and the reason that was claimed as a reaction to that outweigh claim is still open \( Y \), the dialog continues with discussing this reason con. The dialog continues in the same way as after rule 5a.

Rule 16b.

\[
\begin{array}{c}
X \\
\end{array} \quad \begin{array}{c}
Y \\
\end{array} \quad \begin{array}{c}
X \\
\end{array} \quad \begin{array}{c}
Y \\
\end{array}
\]

This paragraph of rule 16 is about when \text{outweighs} \( X \) was already withdrawn, and the counter reason \( Y \) is either withdrawn (left tree) or accepted (right tree). In that case the dialog continues as after rule 5b.
Rule 16c.

First, the outweigh claim (X) is still open, and the counter reason (Y) is not open. In that case the player P is forced to withdraw his outweigh claim if he has accepted the counter reason (right tree). He is forced to withdraw, because the players both are committed to a reason (the counter reason) that yet is not included in outweighs. As a consequence outweighs is no longer appropriately filled, so withdraw is forced (rule 16c1). If the player P is not committed to the counter reason, the dialog continues with a new reaction to the outweigh claim by player P’ (rule 16c2, left tree).

Rule 16d.

If a player accepts an outweigh claim concerning S, he normally is forced to accept S. This rule is about the situation in which the player P’ has accepted the outweigh claim, and is still committed to his reason against S. In that case he must first withdraw that reason. In the move following the forced withdrawal, he is forced to accept S.

7. DiaLaw, the Prolog program

The formal description of the dialog rules is interesting in its own right, but also meant to facilitate the implementation of DiaLaw. In this section the implementation is discussed in main lines. Those interested in precise details of the program code are referred to the appendix.18

Structures about the making of windows or about messages the player receives, are left out. Furthermore, if a structure contains one or more objects, these objects are only shown when this is necessary for the explanation of the code. Otherwise, the number of objects are replaced with a number indicating its arity. For instance, the structure date(Year, Month, Day) would become date/3.

Move, dialog and commitment store

The move, dialog, and commitment store are based on the definitions in section 3. Firstly, the move is a variable DialogMove that refers to a structure with three elements:

(Player, (Act, Sentence), (Level, MoveAbout))

The three elements of this structure have the following properties:

1. the variable Player refers to the player whose turn to move it is, which is either bert (player1) or ernie (player2);

18 The appendix and this section is best understood if one is acquainted with Prolog or a similar declarative language.
2. \((\text{Act, Sentence})\), the variable \text{Act} refers to the illocutionary act performed by the player of the dialog move, the variable \text{Sentence} refers to the propositional content of the illocutionary act;

3. \((\text{Level, MoveAbout})\), the variable \text{Level} refers to the level the dialog move was made on, the variable \text{MoveAbout} refers to the sentence the move is an argument for or reaction to.

Secondly, the dialog is a variable \text{DialogMoves} that refers to the Prolog-list with one or more elements of the type \text{DialogMove}. The number of elements in \text{Dialog} is identical to the number of valid moves of the game. For instance, after three valid moves \text{DialogMoves} is a list with three elements.

Thirdly, the variable \text{CommitmentStore} refers to the Prolog list that contains the commitments of each player at a particular stage of the dialog. Suppose only the first player is committed to \text{murderer(oj)}, and both are committed to \text{shot_wife(oj)}. The commitment store would then be the following list:

\[
\{(\text{bert, murderer(oj)}),
\quad \text{(bert, shot_wife(oj))},
\quad \text{(ernie, shot_wife(oj))}\}.
\]

**The main clause**

Typing \text{`dialaw`} starts the program. It calls the following Prolog-rule, which is the main clause of the program.

\[
dialaw/0 :-
\quad \text{first_move/3},
\quad \text{dialog/3}.
\]

This rule defines under what circumstances \text{dialaw/0} is true, or, in other words, under what circumstances the program ends. This is the case if both \text{first_move/3} and \text{dialog/3} are true.

**On \text{first_move/3}**

Since the first move is the only move without a preceding move, it is dealt with differently from the other moves. The following clause represents the first move:

\[
\text{first_move}(\text{CommitmentStore}, \text{FirstSentence}, \text{DialogMoves})
\]

Each dialog starts with a claim by one of the players, reason why the illocutionary act of the first move is fixed as claim. The only freedom the player of the first move has, is to enter the propositional content of the claim. If he has done this, \text{first_move/3} becomes true. After the first move \text{CommitmentStore} has only one element, viz. the pair that the first player is committed to the sentence he has entered in the first move. The list \text{DialogMoves} also contains only one element, namely that the first player has claimed a particular sentence on the zero level. Because his move is not a reaction to another move the variable \text{MoveAbout} is initiated as \text{dialaw} (see definition 5). Assume Bert claims in the first move the sentence \text{murderer(oj)}. In that case \text{first_move/3} becomes true with the following content:

\[
\text{first_move}(\{(\text{player1, murderer(oj)})),
\quad \text{murderer(oj)},
\quad \{(\text{player1, (claim, murderer(oj))}, \text{(0, dialaw))}))
\]
On dialog/3
The clause dialog/3 has the same elements as that of the first move, namely:

```
  dialog(CommitmentStore, FirstSentence, DialogMoves).
```

After the first move, the program continues until dialog/3 is true. There are basically two situations in which dialog/3 becomes true, so situations in which the dialog ends. First, if after a move the set of open sentences is empty. This is the case if one of the following four clauses is true.

```
  dialog(_, S, [[_, (accept, S), _]|Rest]].
  dialog(_, S, [[_, (withdraw, S), _]|Rest]].
  dialog(CS, S, [[_, (accept, not(S)), _]|Rest]]).
  not_inCS(_, S), CS).
```

After one of the above four moves, the set of open sentences has become empty. If the first sentence is accepted or withdrawn, or if the denial of the first sentence is withdrawn (and the first sentence already was withdrawn), or is accepted. In all these cases the dialog is finished.

Second, if a player has entered quit as an illocutionary act. An unfinished dialog is what remains then.

```
  dialog(_, _, [[_, (quit, _), _]|_]].
```

The quit-option is built in because otherwise the program would keep running in case players do not want to continue an unfinished dialog.

In addition to the five clauses that define the circumstances under which dialog/3 becomes true, there is a sixth dialog/3 clause, namely the following recursive Prolog-rule.

```
  dialog/3 :-
    move_input/5,
    valid_move/6,
    update_CS/5,
    dialog/3.
```

In move_input/5 is determined which player’s turn it is and what the levels are. It is also here that forced moves are dealt with. In case a player is forced to perform some move, he is not asked to enter a move but the move is performed automatically.

If the player whose turn it is has entered a move, the validity of the move is checked (valid_move/6). Note that although the validity of a forced move is checked, it has to be valid always. In case the move appears to be invalid, the player is asked to give in a new move.

Only if the move is valid, the commitment store is updated (update_CS/5). If the commitment store is updated, dialog/3 is started again. This time with a new set of dialog elements (the old list to which one move is added), and the updated commitment store.

An actual dialog
Below is an example of a dialog played with the program. The screenshot represents the sample dialog that will be discussed in Chapter 5, section 2. The upper window, called
status, indicates the status of the move. In the example it is the seventh move, the level is 1, and the move is about `reason(pc(ty), not(not sa(ty)))`. This reason means that because of a probation (pc) condition it was allowed to search (sa) Tyrell,\(^1^9\), or more precisely, it is denied that it was not allowed to search. The next window, called move, is destined for the input of the player. A player gives in first an illocutionary act. After the illocutionary act, the player has to enter the propositional content. If a move is valid, it appears in the window that is called dialog. In the present example the dialog is finished, so there is no possibility to enter new moves. The user is asked whether he wants to leave the screens. Finally, in the lowest windows, called `cs_ernie` and `cs_bert`, after each move the commitment of the players is shown. Since the present dialog is a finished one, at the seventh move the commitment store of Bert and Ernie are identical.

\(^1^9\) See for a further explanation Chapter 5, section 2, and Chapter 4, section 1.
Chapter 4

DiaLaw in action

And you can find a reason for anything
And you can make excuses for all bad things
And you can justify

In this chapter two legal cases are modeled in DiaLaw. The sample dialogs mainly serve two goals. First, the dialogs illustrate the definitions and rules of the previous chapter. Second, the examples show how legal justification can be modeled in DiaLaw.

The chapter is structured as follows. In section 1 a full dialog of 21 moves is presented. This dialog is based on a Californian case concerning (il)legally obtained evidence. In this dialog various aspects of DiaLaw pass review. Second, the Chabot case\(^2\) is used. This case is called after the physician Chabot who carried out euthanasia on a mentally suffering patient. Examples based on the Chabot case show how rules 9-16 work. The dialogs in this section are relatively short.

For convenience sentences are referred to by the number of the move they were claimed in. So a sentence claimed in the third move may thereafter be represented by S\(_3\). The layout of the moves is as follows.

\[ M_n = (\text{Player}, \text{Act}, \text{Sentence}\_1, \text{Level}, \text{Sentence}\_2) \]

On the first line of the move is indicated who the player is, and what type of illocutionary act he performs. The second line presents the sentence that the act is about. Consecutive lines are used in case this is necessary because of the length of the sentence. The last line begins with the level, and ends with the sentence the move is an argument for or reaction to. Note that Sentence\(_1\) and Sentence\(_2\) may be identical.

Between brackets the relevant rules or definitions are indicated, in the following way. A definition is referred to as Def., so definition 1 is referred to as Def. 1; a rule is referred to as R, so rule 1a becomes R1a. If a relevant rule is used in the same way as previously in the dialog, it is not repeated.

1. **The Tyrell Case**

The dialog in this section is based on a case of the Supreme Court of California, the people v. Tyrell. On October 3, 1991, Tyrell attends a football game with two fellow gang members. The week before there had been a shooting incident at a game. The police is afraid that it will happen again and is therefore very vigilant. One of Tyrell's gang

---

1 Lyrics from the 1993 album *Scenes from the second storey* by The God Machine.

members attracts the attention of the police officers, because he is wearing a heavy, quilted coat—although the temperature is in the eighties.\(^3\) They are all searched, and on Tyrell marihuana is found. So far it seems a clear case of illegally obtained evidence, but there is a complicating factor. Namely, Tyrell had been placed on probation subject to, amongst others, the condition: “to submit to a search of his person and property, with or without a warrant, by any law enforcement officer...”. The searching officer, however, is unaware of the probation condition. The question that arises is the following. Is evidence concerning the possession of marihuana obtained illegally, because the search was without probable cause, or was the evidence obtained legally given the probation condition?

Each sequence of moves is preceded by an informal commentary on the dialog, and followed by an explanation of the moves.

One day, Bert and Ernie are discussing the Tyrell case. Bert says that Tyrell is guilty of the possession of an illegal drug, and Ernie fully agrees. This ends their discussion. It is the shortest possible finished dialog, modeled as follows.

\(^{M_3}\) = (Bert, claim, guilty(tyrell), 0, dialaw)

\(^{M_2}\) = (Ernie, accept, guilty(tyrell), 0, guilty(tyrell))

As always the dialog starts with a claim by player1: Bert claims that Tyrell is guilty; the move is on level 0 and is by stipulation about dialaw (Def. 5). After this move Bert is committed to the sentence guilty(tyrell) (R1a). The second move is also on level 0, and it contains a reaction to the sentence that Tyrell is guilty (R3). Namely, Ernie accepts the sentence guilty(tyrell) (R2c). Because Ernie is now also committed to guilty(tyrell) (R1b1), there are no disputed sentences after this second move, \(O_1 = \emptyset\) (see Def. 4). Therefore, the dialog ends (Def. 5).

The next day, after a good night's sleep, Bert wants to be affirmed once more in his opinion about Tyrell's guilt. Bert says: “Well Ernie, Tyrell is guilty, is he not”. This time Ernie is not so easy-going and questions Bert's claim. As an argument Bert claims that Tyrell possessed marihuana. Ernie counter attacks by saying that it is not allowed to claim that Tyrell had marihuana on him. Bert is rather surprised, and questions Ernie's claim.

\(^{M_3}\) = (Bert, claim, guilty(tyrell), 0, dialaw)

\(^{M_2}\) = (Ernie, question, S_1, 0, S_1)

\(^{M_4}\) = (Bert, claim, pos_marihuana(tyrell), l, S_1)

\(^{M_4}\) = (Ernie, claim, illegally_claimed(pos_marihuana(tyrell)), l, S_1)

\(^3\) Obviously the temperature is expressed in Fahrenheit, comparable to 30 °C.
DiaLaw in action

\[ M_5 = (\text{Bert, question, } S_4, 1, S_4) \]

In reaction to guilty(tyrell), Ernie questions in the second move Bert’s claim (R3a). The level in the third move becomes lower, because of the questioning in the second move (R4). After Bert claimed that Tyrell possessed marihuana, in the fourth move Ernie claims that Bert’s claim was illegal (R7). Bert questions Ernie’s claim.

In the dialog tree of the first five moves, DT_5, each sentence is represented by the conventional abbreviation.

Ernie gave the case a second thought. He no longer thinks Tyrell is guilty. He assumes that it is not allowed to adduce that Tyrell possessed marihuana, because the knowledge of this fact is derived from illegally obtained evidence. Bert questions this claim. Ernie starts his defense by claiming the validity of a rule that says “a fact is illegally claimed, if it is based on illegal evidence”. Bert agrees to that rule. When Ernie continues to say that the evidence was in fact illegally obtained, Bert wants to know why this is the case.

\[ M_6 = (\text{Ernie, claim, reason(legally_ob_ev(pos_marihuana(tyrell)), } S_4), 2, S_4) \]

\[ M_7 = (\text{Bert, question, } S_4, 2, S_4) \]

\[ M_8 = (\text{Ernie, claim, valid(rule(legally_ob_ev(Fact), illegally_claimed(Fact))), } 3, S_4) \]

\[ M_9 = (\text{Bert, accept, } S_4, 3, S_4) \]

\[ M_{10} = (\text{Ernie, claim, illegally_ob_ev(pos_marihuana(tyrell)), } 3, S_4) \]

\[ M_{11} = (\text{Bert, question, } S_{10}, 3, S_{10}) \]

In the sixth move Ernie claims that the fact that the evidence was illegally obtained, is a reason why the claim stating that Tyrell possessed marihuana is illegal (R11). In the eighth move, the level has become lower once more. Ernie claims the validity of a rule, which is accepted by Bert. Still on level 3, Ernie claims that the condition of the rule is satisfied. Bert questions this sentence. The following picture shows the dialog tree after eleven moves.
Bert and Ernie continue their discussion. Ernie says: “Listen Bert, don’t you think that the fact that there was no probable cause concerning Tyrell’s case, since he was not suspected of illegal drug possession, is a reason why the evidence is obtained illegally”. Bert admits this and makes Ernie happy, because Ernie believes he has won the dispute about the evidence. But then Bert reminds Ernie that because of Tyrell’s probation condition, he had to allow a search at any time and for whatever reasons or no reasons at all. Ernie agrees, but still believes that his argument is stronger than Bert’s.

\[
M_{12} = (\text{Ernie}, \text{claim}, \\
\quad \text{reason}(-\text{suspect(tyrell)}, \\
\quad \quad \text{illegally_ob_ev(pos_marihuana(tyrell))}, \\
\quad 4, S_8)
\]

\[
M_{13} = (\text{Bert}, \text{accept}, \\
\quad S_{12}, \\
\quad 4, S_8)
\]

\[
M_{14} = (\text{Ernie}, \text{claim}, \\
\quad \text{outweighs}((-\text{suspect(tyrell)}), \{\}, S_{10}), \\
\quad 4, S_8)
\]

\[
M_{15} = (\text{Bert}, \text{claim}, \\
\quad \text{reason(prob_cond(tyrell), -S_{10})}, \\
\quad 4, S_{14})
\]

\[
M_{16} = (\text{Ernie}, \text{accept}, \\
\quad S_{15}, \\
\quad 4, S_{14})
\]

\[
M_{17} = (\text{Ernie}, \text{withdraw}, \\
\quad S_{14}, \\
\quad 4, S_8)
\]

\[
M_{18} = (\text{Ernie}, \text{claim}, \\
\quad \text{outweighs}((-\text{suspect(tyrell)}), \{\text{prob_cond(tyrell)}\}, S_{10}), \\
\quad 4, S_8)
\]

Ernie claims a reason, which is accepted by Bert. Next Ernie claims that this reason outweighs the empty set of reasons (R11, R12). This claim of Ernie is decisive (Bert may not question, R13), unless Bert knows a reason con. Bert wants to avoid commitment to $S_{18}$, and therefore claims a reason against $S_{10}$ (R15). Ernie accepts this reason, and as a
consequence he is forced to withdraw 'outweighs' (R16c). After this forced withdrawal, Ernie claims a new 'outweighs', with both reasons in it. After eighteen moves the dialog tree looks as follows.

\[
\begin{array}{c}
\text{0} \quad S_1 \\
\text{1} \quad S_3 \quad S_4 \\
\text{2} \quad S_6 \\
\text{3} \quad S_8 \\
\text{4} \quad S_{12} \quad S_{14} \quad S_{15} \quad S_{18}
\end{array}
\]

Bert wants to know why Ernie believes his argument is stronger than Bert's. Ernie cites a similar case by the Court of Appeal of California. Bert is flabbergasted by this argument, and mumbles: "You really got me Ernie, I withdraw my initial claim about Tyrell's guilt."

\[
M_{19} = (\text{Bert}, \text{question}, \ S_{19}, \quad 4, \ S_{18})
\]

\[
M_{20} = (\text{Ernie}, \text{claim},
\text{reason(Cal\_court(outweighs([-suspect(P)]),}
\quad \{\text{prob\_cond}(P), \text{illegally\_ob\_ev}(P)\}), \ S_{18}),
\quad 5, \ S_{18})
\]

\[
M_{21} = (\text{Bert}, \text{withdraw},
\ S_1, \quad 5, \ S_{20})
\]

After Bert's question, Ernie adduces a reason that states that in a verdict of the Court of Appeal of California in a similar case the outcome was the same as Ernie thinks it should be for Tyrell. Although Bert is not forced to do so, he finds the reason convincing enough to withdraw his initial claim. After this withdrawal there are no open sentences left, so the dialog ends. After the last, 21st move, there are no disputed sentences left, as the following tree shows.
Probably most lawyers will agree with the decision of the Court of Appeal. Surprisingly, the Supreme Court of California did not. They decided that the evidence was obtained legally. So, if Bert would have been aware of this verdict, he might have won the dialog.

2. The Chabot Case

On June 21, 1994, the Dutch Supreme Court decided a case of euthanasia which not only had impact on legal doctrine, but also attracted a lot of attention in the media. The case was special, because for the first time euthanasia was deemed permissible for a merely mentally suffering patient. Section 294 from the Dutch Penal code says, as far as relevant here, freely translated: \(^4\)

Someone who intentionally helps another to commit suicide is punishable.

A physician is obligated to report on each case of euthanasia he carries out. If his decision has been careful, according to relevant guidelines, he will not be prosecuted. The legal ground for not prosecuting is force majeure, \(^5\) i.e., the existence of two contrary duties. On the one hand the duty to preserve life, and, on the other hand, the duty to respect a person and his explicit wish to die.

\(^4\) In Dutch: Art. 294 Sr. Hij die opzettelijk een ander tot zelfmoord aanzet, hem daarbij opzettelijk behulpzaam is of hem middelen daartoe verschaf, wordt, indien de zelfmoord volgt, gestraft (...)

\(^5\) This is the material legal ground. Procedurally, the prosecutor applies the so-called opportunity principle: the right to drop a charge if the decision to drop is based on grounds of public interest, section 167 of Dutch Code of Criminal Procedure (art. 167 Sv).
In section 2.1, 2.2, and 2.3 examples are given of the rules that were introduced in Chapter 3, section 6.2, 6.3, and 6.4, respectively. The following two moves precede the dialogs of the next sections.

\( M_1 = \) (Ernie, claim,
            punishable(chabot),
            0, dialaw)
\( M_2 = \) (Bert, question,
            S1,
            0, S1)

In the first move Ernie claims that Chabot, the physician who carried out euthanasia, is punishable. Bert questions this claim.

### 2.1 On reasons to apply a rule

In the dialog below Ernie claims an argument for the sentence that Chabot is punishable. He claims that a particular rule applies. The condition of the rule is that Chabot helped with suicide, the conclusion that Chabot is punishable. Bert questions the claim.

\( M_3 = \) (Ernie, claim,
            applies(rule(help_suicide(chabot), punishable(chabot)),
            1, S1)
\( M_4 = \) (Bert, question,
            S3,
            1, S3)

If a player does not accept that a rule applies, this player has to be convinced by supporting arguments. A good strategy is to claim that

1. the rule is valid, and
2. that the condition of the rule is satisfied.

Together both sentences yield one reason why the rule applies, namely the reason that the rule is applicable (see Chapter 3, section 5)

If the opponent does not accept that the rule is valid, reference to the relevant section probably suffices to back the claim, e.g.,

\( \text{source (rule (help_suicide(X), punishable(X)), Sr_294).} \)

This sentence says that the relevant rule can be found in section 294 of the Dutch Criminal Code, commonly abbreviated as 'Sr'. After the claim of the source, either the player will accept the validity of the rule, or he starts a discussion about why he thinks the rule is not valid. For instance, because the representation of the rule is inaccurate, or the rule was abrogated.

The depth of the justification depends on the questioning of the opponent. In case sentences are accepted they do not have to be further justified, because the fact that sentences are accepted makes that they are justified. Since the purpose of this section is to illustrate rule 9, in the following dialog Bert questions neither the validity of the rule, nor the fact that the condition is satisfied.

\( M_5 = \) (Ernie, claim,
            valid(rule(help_suicide(Person), punishable(Person)),
            2, S3)
\( M_6 = \{\text{Bert, accept,} \)
\( \quad S_s, \)
\( \quad 2, S_s\}\)
\( M_7 = \{\text{Ernie, claim,} \)
\( \quad \text{help_suicide(chabot),} \)
\( \quad 2, S_s\}\)
\( M_8 = \{\text{Bert, accept,} \)
\( \quad S_s, \)
\( \quad 2, S_s\}\)
\( M_9 = \{\text{Ernie, claim,} \)
\( \quad \text{reason:} \)
\( \quad \text{applicable(rule(help_suicide(chabot), punishable(chabot))),} \)
\( \quad \text{applies(rule(help_suicide(chabot), punishable(chabot))),} \)
\( \quad 2, S_s\}\)

At this stage of the dialog Bert is committed to both that the rule is valid \((S_s)\), and that the condition of the rule is satisfied \((S_s)\). These two facts yield one reason to apply a rule. This is the reason Ernie claimed in the ninth move, namely that the rule is applicable. Because of Bert’s commitment and the fact that the reason that the rule is applicable is open, Bert is forced by rule 9a to accept this reason in the tenth move.

\( M_{10} = \{\text{Bert, accept,} \)
\( \quad S_s, \)
\( \quad 2, S_s\}\)

Suppose that in the sixth move Bert had denied that the rule is valid, and Ernie became convinced that indeed the rule is not valid.

\( M_{11} = \{\text{Ernie, accept,} \)
\( \quad \text{\neg valid(rule(help_suicide(Person), punishable(Person))),} \)
\( \quad \neg, \neg\}\)

After this move \( M_n \), rule 9b forbids the claim of the reason that the rule is applicable. The same holds if Ernie would be committed to the negation of the condition of the rule, so to \( \text{\neg help_suicide(chabot)} \).

2.2 On applies, reason, and excluded

This section focuses on the relation between applies, reason, and excluded \((R10)\). After Bert’s questioning of the claim that Chabot is punishable \((move 2)\), a supporting argument could be given different from the one of the previous section. Namely, that the fact that Chabot assisted someone in committing suicide is a reason why he is punishable. If Bert questions this reason, Ernie can support it by claiming that the related rule applies.

\( M_1 = \{\text{Ernie, claim,} \)
\( \quad \text{reason((help_suicide(chabot), punishable(chabot)),} \)
\( \quad 1, S_s\}\)
\( M_2 = \{\text{Bert, question,} \)
\( \quad S_s, \)
\( \quad 1, S_s\}\)
\( M_3 = \{\text{Ernie, claim,} \)
\( \quad \text{applies(rule(help_suicide(chabot), punishable(chabot))),} \)
\( \quad 2, S_s\}\)
DiaLaw in action

\[ M_4 = \{ \text{Bert, accept,} \]
\[ \quad S_5, \]
\[ \quad 2, S_6 \} \]

In the previous section it was elaborated how the dialog can progress if \text{applies} is questioned. Here, Bert accepts that the rule applies, so he is committed to that sentence. The reason related to the applied rule is still open, so in the seventh move Bert is forced by rule 10a to accept the reason claimed in the third move.

\[ M_4 = \{ \text{Bert, accept,} \]
\[ \quad \text{reason}((\text{help\_suicide(chabot)}, \text{punishable(chabot)}), \]
\[ \quad 2, S_6) \} \]

The reason for this forced acceptance is that if a rule applies the related reason originates (Chapter 3, section 5).

Assume, Bert did not accept in move 6 that the rule applies, but denied it and Ernie questioned this denial. As an argument for the negation of applies, Bert can adduce that the rule is excluded. The next section elaborates on which arguments can be used to support the claim that the rule is excluded. For now, let us assume that Ernie immediately accepts that the rule is excluded.

\[ M_6 = \{ \text{Bert, claim,} \]
\[ \quad \neg S_5, \]
\[ \quad 2, S_6 \} \]

\[ M_7 = \{ \text{Ernie, question,} \]
\[ \quad S_6, \]
\[ \quad 2, S_6 \} \]

\[ M_8 = \{ \text{Bert, claim,} \]
\[ \quad \text{excluded}(\text{rule}(\text{help\_suicide(chabot)}, \text{punishable(chabot)}), \]
\[ \quad 2, S_6) \} \]

\[ M_9 = \{ \text{Ernie, accept,} \]
\[ \quad S_8, \]
\[ \quad 2, S_8 \} \]

After this dialog Ernie is committed to both that the rule applies and that the rule is excluded. Because commitment to these sentences cannot go together, Ernie is forced by rule 10b to withdraw the sentence that the rule applies.

\[ M_{10} = \{ \text{Ernie, withdraw,} \]
\[ \quad \text{applies}(\text{rule}(\text{help\_suicide(chabot)}, \text{punishable(chabot)}), \]
\[ \quad 2, S_9) \} \]

\[ 2.3 \text{ On reasons and outweighs} \]

This section gives examples of reason and outweighs, rule 11-16. In the previous subsection, Bert claimed in the eighth move that the rule Ernie wanted to apply is excluded. Ernie accepted this above, but he now questions the claim that the rule is excluded. As an argument Bert puts forward that Chabot acted in force majeure. Ernie questions this claim. Bert adduces as a reason the death wish of Hilly B.\(^6\) the woman who committed suicide.

\(^6\) Apparently Bert knows that the woman Chabot carried out euthanasia on is Hilly B.
Ernie accepts this reason. Bert claims that this single reason outweighs an empty set of reasons.

\[ M_9 = \langle \text{Ernie, question}, \text{excluded(\text{rule(\text{help_suicide(chabot)}, \text{punishable(chabot))}, 2, S_9)} \rangle \]

\[ M_{10} = \langle \text{Bert, claim, force_majeure(chabot)} \rangle \]

\[ M_{11} = \langle \text{Ernie, question, S_{10}, 3, S_{10}} \rangle \]

\[ M_{12} = \langle \text{Bert, claim, reason(death_wish(hillary_b), force_majeure(chabot))}, 4, S_{10} \rangle \]

\[ M_{13} = \langle \text{Ernie, accept, S_{11}, 4, S_{11}} \rangle \]

\[ M_{14} = \langle \text{Bert, claim, outweighs(\{death_wish(hillary_b), \}, force_majeure(chabot))}, 4, S_{10} \rangle \]

In the twelfth move Bert claims a reason. According to rule 11 a reason may be claimed on a certain level, if one level higher the sentence was claimed that it is a reason for. Here the reason is claimed on level 4, and it is a reason for the force majeure of Chabot. Since the force majeure of Chabot was claimed on level 3, the claim of the reason in move 12 is allowed.

In the fourteenth move Bert claims an outweigh sentence on level 4 that pleads for the force majeure of Chabot. Since the force majeure of Chabot was claimed on level 3, this claim is allowed by rule 11. Rule 12 prescribes that the set of reasons pro may not be empty, and that the sentence outweighs pleads for is still disputed. Since there is no agreement yet about the force majeure of Chabot, and there is one reason in the set of reasons pro, these conditions are also fulfilled.

The dialog continues as follows. Ernie claims a counter reason, namely that life should be respected. Bert agrees that the respect for life is a reason against the force majeure of Chabot.

\[ M_{15} = \langle \text{Ernie, claim, reason(respect_life, force_majeure(chabot))}, 4, S_{11} \rangle \]

\[ M_{16} = \langle \text{Bert, accept, S_{15}, 4, S_{15}} \rangle \]

Questioning outweigh is forbidden if the set of reasons con is empty (R13). So Ernie was not allowed to question in the fifteenth move. Moreover, by the same rule deny was not allowed either. The fifteenth move by Bert is a special one. Rule 15 allows that in reaction to an outweigh claim a counter reason is claimed. In the sixteenth move Bert accepts the
counter reason. Because of the acceptation of this reason, out weighs is no longer accurate. Therefore, Bert is forced to withdraw out weighs in the seventeenth move (R16c).

\[ M_{17} = (\text{Bert, withdraw, outweights}\{\text{death_wish(hilly_b)}, \}, \text{force_majeure(chabot)}, 4, S_{16}) \]

Rule 16c1 applied here, because:
- The sentence out weigh is still open;
- Bert is committed to the counter reason.

Suppose Ernie had not claimed a counter reason in the fifteenth move, but accepted the out weigh claim.

\[ M_{15} = (\text{Ernie, accept, outweights}\{\text{death_wish(hilly_b)}, \}, \text{force_majeure(chabot)}, 4, S_{15}) \]

Rule 14 deals with the consequences of accepting out weighs. After this fifteenth move, Ernie is forced to accept the sentence out weighs pleaded for, namely that Chabot acted in force majeure.

\[ M_{16} = (\text{Ernie, accept, force_majeure(chabot)}, 4, S_{15}) \]

Rule 14a has as an additional condition that Ernie is not committed to the negation of Chabot’s force majeure, which Ernie is not. If he were, he would first have to withdraw this negation, before being forced to accept (R14b).

3. CONCLUDING REMARKS

Most of the definitions and rules of Chapter 3 were illustrated in this chapter. The way legal justification is modeled in DiaLaw must be clear now, at least in main lines. The definitions and rules will not be changed or extended in the remainder of this thesis.

Only after the chapter was called DialLaw in action, I discovered that Law in action is a well-known concept in Jurisprudence. Pound [1931] introduced the distinction between law in the books and law in action. The division was meant to indicate an alternative direction of research; instead of studying the literature in their ivory tower, academics had to study law in practice.

My information is from the books, and is not obtained from empirical research. Nevertheless, two real legal cases were analyzed in DiaLaw, so in a sense this chapter demonstrated a little law in action.

---

7 Bert accepted a reason. As a consequence the sets of reasons of out weighs are no longer filled with the reasons the players agreed upon. Therefore out weighs must be withdrawn.
Chapter 5

Dialogical models of argumentation

*Nothing compares, nothing compares to ...*¹

DiaLaw is one in a field of many dialogical models of argumentation. This chapter locates DiaLaw in this field.² First, in section 1, of a wide range of dialogical models is briefly indicated for which purpose they were developed.

The core of the chapter consists of a discussion of the following work:

- MacKenzie’s DC (section 0);
- Rescher’s Dialectics and the formal elaboration by Brewka (section 0);
- Gordon’s Pleadings Game (section 0);

The reason to choose these games is that they are open models³ like DiaLaw: no fixed sets of premises, statements or arguments are used. The games define the structure and sequences of moves; the players can freely choose the content of each move.

To introduce the models, the same dialog is represented in MacKenzie’s DC, in Rescher’s Dialectics, in Gordon’s Pleadings Game, and in DiaLaw (section 0). In the discussion of each of these models⁴ special attention is paid to the possible (sequences of) moves and commitment. Section 0 provides a survey of moves and commitment of DiaLaw and the models discussed in sections 0-0.

In the last section other contributions to the field of dialogical argumentation are discussed: the work of the Dutch linguists Van Eemeren & Grootendorst, Vreeswijk’s dialectics, the HELIC system by Nitta and colleagues, and Farley & Freeman’s levels of burden of proof.

1. The purpose of models

Each model of argumentation is developed with a particular goal in mind. Although on a general level all models share the aim to model argumentation, their purposes vary on a less general level.

Hamblin [1970, p. 265f.] developed the dialog game H to investigate fallacies. MacKenzie’s games [1979a, 1990] are a further development of H, primarily meant to block the use of circles in argumentation. Van Eemeren & Grootendorst [1982] wanted their model to be used both as an instrument for analyzing discussions, and as a code of

---

¹ Lyrics from the 1990 album *I do not want what I have not got* by Sinéad O’Connor.

² For a general overview of dialogical models see the traditional Barth & Krabbe [1982], Walton & Krabbe [1995], and Van Eemeren *et al.* [1996].

³ See Chapter 2.

⁴ For a discussion of DiaLaw, see Chapter 3.
behavior for reasonable discussions. All these models have in common that they are language-oriented, meant to model practical reasoning.

The purpose of the AI & Law model the Pleadings Game [Gordon, 1995] is to identify on what points parties in a civil lawsuit do not agree, or, in legal terms, to determine what issues -both legal and factual- exist between the parties. The HELIC-systems [Nitta, Wong & Ohtake, 1993; Nitta et al., 1995; Nitta & Shibasaki, 1997] aim at developing a unified model of legal reasoning and also a portable software tool based on this model. The purpose of DiaLaw\(^5\) is to model legal justification. The model is not intended to reflect a specific domain, but aims at law in general.

The philosopher Rescher [1977] wanted to investigate how knowledge could be improved through dialectics. He thought rationality in academic dispute could be guaranteed by a dialectic exchange of arguments. In initiating the current popularity of dialectics, Loui has played a major role. Loui [e.g., 1992] stresses unreservedly that argumentation is a process. The purpose of his dialectical protocols [Loui, 1992; Loui et al., 1992; Loui & Chen, 1992] is to provide a step in the direction of fair and effective protocols for argumentation. An important element of his work is the study of rationales of rules and cases [Loui et al., 1993; Loui & Norman, 1995]. The purpose of the dialectics by Vreeswijk [1993] is to test whether some thesis is in force given a particular background. His debating technique is used to search a vast data base of arguments efficiently.

Dialog games can have various other purposes. In the Middle Ages the Obligation Game was used to test the knowledge of students. If during a dialog with their teacher the students were capable to avert contradiction, they passed their exam. Lorenz [1961] designed a dialog game in which the proponent has the burden of proving that his initial position is a tautology. Bench-Capon and colleagues [Bench-Capon, Lowes & McEnery, 1991; Bench-Capon, Dunne & Leng, 1992] explored dialog games to improve the way knowledge-based systems provide explanations. The user is not given a standard explanation, but the explanation is determined by the questions of the user. St-Vincent & Poulin [1994] used interaction between groups of agents to determine (on the basis of majority of agents, unanimity, etc.) the meaning of vague legal concepts. Prakken & Sartor [1996] used a dialog game to determine if an argument is justified in their argumentation theory.

2. A SHORT SAMPLE DIALOG

In all games, that is MacKenzie's DC, Rescher's Dialectics, Gordon's Pleadings Game, and DiaLaw, there are two parties. In the following discussion the participants of the dialog are named as in the original work. The dialog that is used is a shorter version of the Tyrell dialog from the previous chapter, and focuses on two arguments:

- Only suspects may be searched, and Tyrell was not a suspect;
- One of Tyrell's probation conditions was that he had to allow search any time.

Bert: It was not allowed to search Tyrell.
Ernie: Why do you think so?
Bert: Only if someone is a suspect he may be searched, and Tyrell was not a suspect.
Ernie: I agree, but Tyrell was on probation and had to allow a search at any time.

---

\(^5\) See Chapter 2, 3, and 4.
Bert: You are right, search was allowed.

Although there are more ways this dialog could have been modeled, I believe the forthcoming representations reflect the spirit of the original work. It is not intended that the models are entirely understood by only reading the examples, rather, the dialogs mean to give an impression of the games. In subsequent sections specific details of the games are elaborated. In the sample dialogs the following abbreviations are used:

- sa = search was allowed;
- s = suspect;
- pc = probation condition.

**MacKenzie’s DC**

<table>
<thead>
<tr>
<th></th>
<th>Wilma</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>~sa</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>~s</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>sa</td>
<td></td>
</tr>
</tbody>
</table>

In the second move Bob challenges the statement Wilma started with. In defense Wilma puts forward the statement that there was not a suspect. Bob replies with the statement that if the probation condition is fulfilled, search is allowed. On the basis of this information Wilma decides that search was allowed after all.

**Rescher’s Dialectics**

<table>
<thead>
<tr>
<th></th>
<th>PRO</th>
<th>CON</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>!→sa</td>
<td>→sa</td>
</tr>
<tr>
<td>2.</td>
<td>¬sa/¬s &amp; !¬s</td>
<td>sa/(¬s &amp; pc) &amp; !¬s &amp; pc</td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The dialog starts at the moment the proponent categorically asserts that search was not allowed. The opponent replies with the cautious assertion that search was allowed. In reply the proponent asserts that someone who is not a suspect, may not be searched, and that there was no suspect. The opponent replies that in case someone is not a suspect, but subject to this specific probation condition, search is allowed. Since the proponent cannot answer the last move of the opponent, the dialog stops and is won by the opponent.

**Gordon’s Pleadings Game**

<table>
<thead>
<tr>
<th>Plaintiff</th>
<th>Defendant</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td></td>
</tr>
</tbody>
</table>
| 2.        | declare(rule, 1, (X)
|           | if (¬s X)
|           | then (¬sa X))
|           | defend(denial(claim(¬sa ty)))
|           | { ¬s ty
|           | backing l
|           | op(inst 1(parms ty))
|           | )}        |
concede(claim (backing 1))
concede(claim (~s ty))
declare(rule, 2, (X)
  if (pc X)
  then (sa X))
defend({argument (~s ty)
  backing 1
  ap(inst 1(parms ty))
  ~sa ty)
  ( pc ty
  backing 2
  ap(inst 2(parms ty)))
}

concede(claim (backing 2))
concede(claim (pc ty))
concede(rebuttal (pc ty)
  (sa ty) (~s ty))

After the plaintiff has filed that the search was illegal, the dialog starts with the denial of that claim. In the second move the plaintiff declares a rule, and puts forward an argument based on this rule, that defeats the denial. In his reaction the defendant concedes the validity of the rule, also concedes that the condition of the rule is satisfied, and declares a new rule. He applies this new rule to defeat the argument of the plaintiff. In the last move the plaintiff concedes all moves left open, including the rebuttal, so he loses the debate.

**DiaLaw**

<table>
<thead>
<tr>
<th>Bert</th>
<th>Ernie</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. claim, ~sa(ty)</td>
<td>question, ~sa(ty)</td>
</tr>
<tr>
<td>2.</td>
<td></td>
</tr>
<tr>
<td>3. claim, reason(~s(ty), ~sa(ty))</td>
<td>accept, reason(~s(ty), ~sa(ty))</td>
</tr>
<tr>
<td>4.</td>
<td></td>
</tr>
<tr>
<td>5. claim, outweighs(~s(ty), {}, ~sa(ty))</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td></td>
</tr>
<tr>
<td>7. withdraw, ~sa(ty)</td>
<td>claim, reason(pc(ty), sa(ty))</td>
</tr>
</tbody>
</table>

The moves 1-6 are similar to move 10-15 of the Tyrell dialog in Chapter 4, except that the reasons in this dialog are not about the illegality of the obtained evidence. In the seventh move Bert withdraws the claim made in the first move.

Some differences between the models are already apparent. For instance, that defeat is based on specificity in Rescher’s game, that Gordon’s arguments are based on rule application, that contradiction is allowed in MacKenzie’s game, and that in DiaLaw reasons are weighed.
3. **MacKenzie’s DC and Hamblin’s H**

To investigate fallacies, Hamblin [1970, p. 265f.] developed the dialog game ‘Why-Because system with questions’, often abbreviated as ‘H’. To my knowledge, the game H was the first to use the notion of commitment in dialogs and the possibility to retract moves. Moreover, Hamblin invented the term *formal dialectic* and was the first to define a system of rules for exchanging arguments in a rational discussion [cf. Van Eemeren, Grootendorst & Kruijger, 1987, p. 136].

A well-known variant of the game H is the game DC by MacKenzie. Because much other research [e.g., Finkelstein & Fuks, 1990; Bench-Capon, Lowes & McEnery, 1991; Pilkington et al., 1992; Moore, 1993] is based on the game DC, I will discuss the game DC rather than H. The discussion focuses especially on the types of moves and commitment.

**The moves**

The players perform moves in turn, and are only allowed to make one locution at each turn. A locution in DC is comparable to the illocutionary act and its propositional content in DiaLaw. There are five different locutions or move types.\(^6\)

<table>
<thead>
<tr>
<th>Move type</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Statement</td>
<td>‘P’, ‘Q’, etc. (and ‘Not P’, ‘If P then Q’, and ‘Both P and Q’)</td>
</tr>
<tr>
<td>2. Withdrawal</td>
<td>‘No commitment P’</td>
</tr>
<tr>
<td>3. Question</td>
<td>‘Is it the case that P?’</td>
</tr>
<tr>
<td>4. Challenge</td>
<td>‘Why P?’</td>
</tr>
<tr>
<td>5. Resolution demand</td>
<td>‘Resolve whether P’</td>
</tr>
</tbody>
</table>

**Statements**

The move Statement is used for the introduction of statements. In addition to normal statements, also the ordinary compounds of statements may be claimed. After the move statement, both players are committed to the statement. The principle *silence implies consent* is used, because it saves time (not every statement a player agrees to has to be conceded) and it fits in with daily life discussions (normally someone will let it be known if he disagrees). The principle structures the dialog: it forces the players to react to what the other says, in order to prevent commitment to statements they do not want to be committed to.

The move following a statement can be any of the five moves, including statement.

**Withdrawals**

The move Withdrawal is used for the retraction of statements. By this move the commitment of the moving player ends. In [MacKenzie, 1990] the withdrawal is

---

\(^6\) Gordon [1995, p. 107] mentions MacKenzie as the first who used commitment stores and retraction of moves. However, MacKenzie [1979a] ascribes these novelties to Hamblin.

\(^7\) Actually MacKenzie calls moves locutions (like Hamblin), but for the sake of uniformity I refer to them as moves.
represented differently, namely by 'I'm not sure that P'. This alternative representation emphasizes the nature of commitment. In principle someone is committed to what he is sure about, to what he believes. If there are doubts about a statement, this normally means that a player wants to hear (additional) arguments from the other player. The opponent can be asked to give arguments with the move challenge (see below).

Withdrawals are not allowed in case the statement is a "logicians' conditional". For instance, in case the Modus Ponens argument is used, the statement 'If both P and If P then Q, then Q' is a logicians' conditional. For obvious reasons these conditionals may neither be denied, nor challenged.⁸

Again, the move following a withdrawal can be any move.

Questions
A question (is it the case that P?) forces the opponent to make his position about the statement P clear. There are no preconditions for this move, what makes it possible for instance to ask this question to a player who is already committed to P. Although this appears to be redundant, a player may sometimes want to be sure. For instance, if a player introduces the material implication 'If P then Q' in order to prove Q, he may want to be sure about the opponent's position regarding P.

The move question has no consequences for commitment.

The other player can react to a question in three different ways. He can confirm the statement by claiming it (P). He can deny the statement by claiming its negation (not P). Finally, he can refrain himself from saying anything at all about the status of P (No commitment P). If a player already was committed to P, the consequence of this last reaction is that he now no longer is.

Challenges
The challenge is a demand for evidence for a particular statement. The consequence of a challenge is that the player who challenges, in case he was committed, loses his commitment.⁹ Moreover, the challenged player becomes committed. This may seem a more severe consequence than it actually is, because the challenged player can withdraw the statement in the next move. He may, for example, want to withdraw the statement if he has no evidence to support it. A final consequence for commitment is that the player who challenges, becomes committed to the challenge (Why P?). The commitment to a challenge cannot be ended. This commitment to 'Why P?' is used throughout the game to block question begging: it is not allowed to answer 'Why P?' with P or any other statement challenged previously. So, if 'Why P?' is part of the commitment store, P may not be claimed in answer to a challenge.¹⁰ By this rule also remote circles are blocked, e.g., the last move of the following sequence is forbidden:

---

⁸ See also MacKenzie's (1979b).
⁹ Instead of the automatic loss of commitment, Hamblin allowed both withdrawal and challenge in one turn. This exception to the rule that a turn may only contain one locution is unnecessary in MacKenzie's game.
¹⁰ It seems strange that a player is committed to a challenge. More obvious would be to use the game record of moves for this purpose. It is not clear to me why the record of the game is not used here.
Dialogical models of argumentation

Why P1?
P2
Why P2?
P3
(...)
Why Pn?
P1

If in response to ‘Why P?’ the player claims Q, both players become committed to Q and Q → P. A second possible reaction to a challenge is a withdrawal of the challenged statement. Finally, an answer to a challenge can be a resolution demand in which the consequent is the challenged statement, and the condition is a statement the other player is committed to.

Wilma: Why P?
Bob: Resolve whether Q → P

This reaction of Bob is possible only if Wilma was already committed to Q and Q → P. This move comes down to: “How can you ask ‘Why P?’ if you are committed both Q and Q → P?”

Resolution demands

This move is meant to confront the opponent with an inconsistency in his commitment store. There are two different resolution demands. The first type is a demand to resolve an inconsistency in one’s commitment store. For instance, resolve whether Q & ¬Q, or, resolve whether (P & (P → Q)) & ¬Q. The only reaction allowed to this resolution demand is the withdrawal of one of the contradicting conjuncts.

The second type of resolution demand is the one discussed under challenges: the move ‘resolve whether Q → P’, after P is challenged. Note that this resolution demand is only allowed if the opponent is committed to both Q and Q → P. The reaction to this resolution demand can be the affirmation of the consequent (P), or the withdrawal of either Q or Q → P.

3.2 Commitment

The commitment stores in DC are empty at the beginning of the dialog. This is different from Hamblin [1970, p. 265], who included the axioms of the language in the initial commitment store. However, in a later writing [Hamblin 1987, p. 240] the commitment store was only empty at the beginning if there had been no prior dialogs between the same parties. If there had been prior dialogs between the parties, information from these was included in the commitment store of the new dialog. This is an attractive idea, because in that case not all dialogs have to start from scratch. If a dialog game is implemented in the computer, storing of prior commitment is not difficult.

In MacKenzie’s game each player has his own commitment store, in which the statements are stored that he is committed to. DiaLaw has an almost identical way of

A possible reason that the challenge is stored as a commitment could be that in the game commitment is by far the most important notion.
storing commitment, except that there is a single commitment store and for each statement it is indicated which player is committed. A small disadvantage of having two commitment stores is that operations concerning commitment have to be performed on two sets instead of one. For instance, if nothing changes in the commitments MacKenzie needs two equations, one for each player (a and b), to express this, \( C_{n+1}(a) = C_n(a) \); and \( C_{n+1}(b) = C_n(b) \). In DiaLaw a simple equation suffices, \( C_{n+1} = C_n \). An advantage of having two commitment stores is that only statements are stored, and not pairs containing the player as well as the statement. A strong point of both approaches is that it is clear at once what the commitment of either player is.

The commitment store is not closed under logical consequence. It is for instance possible to be committed to \( P \rightarrow Q \), \( P \) and \( \neg Q \). Only immediate contradiction, e.g., \( P \) and not \( P \), is forbidden. The idea behind this way of modeling commitment is that a player is not omniscient and cannot be aware of all consequences of his commitments, especially not when these consequences are remote. In one of the systems\(^{11}\) discussed in Hamblin [1971] commitment is closed under logical consequence, and it appears that such strict commitment is not very useful in modeling (daily life) discussions.

3.3 Concluding remarks

The purpose of the game DC is modeling daily life discussions, and preventing fallacies, especially circles. If the participants of the game use natural language, the resulting dialog is not a strange sequence of moves, but a representation of the discussion in a natural way. Even if the rules are implemented in a computer [Moore, 1993], dialogs between two human players remain rather natural. This is strong point of the game DC.

The way circles are blocked is not waterproof. If a proposition is claimed that is not identical but synonymous to a previous one, the circle is not recognized. For instance, if a player challenged that someone is murderer, his opponent can try to justify this by claiming that he is a legally punishable killer.\(^{12}\) In a similar way contradictions may not be recognized. Although murder and death by negligence exclude each other, they are not recognized as contradicting. In order to recognize these contradictions, domain knowledge is required. But even if domain knowledge is added, not all circles and contradictions can be recognized. This problem also plays a role in DiaLaw. However, if in DiaLaw a synonym is accepted, it is not necessarily as a reason for the statement it is a synonym of. In DC automatically a material implication is added, e.g., if \( A' \) then \( A \), where \( A' \) is a synonym of \( A \).

When a contradiction occurs in DiaLaw, a discussion can start in which the other player tries to explain why for instance death by negligence and murder exclude each other. In DC such a discussion may also arise, but the automatic concessions cause the following problem. Either this player is committed to death by negligence all the time that he is explaining that it cannot be a case of death by negligence, or he withdraws the statement about death by negligence before starting his explanation. In the latter case his opponent

\(^{11}\) It is system 7 in the paper (the systems are numbered 1a, 1b, 2, 5, 7).

\(^{12}\) For instance, the following dialog contains an unrecognizable circle (where \( A \) is a synonym of \( A' \)):

Bob: A; Wilma: Why A?; Bob: A'.
may claim in the move following the withdrawal of death by negligence, another statement that needs a reaction.

The fact that the rules are material implications makes the model less suitable for legal justification (see Chapter 2).

4. RESCHER’S DIALECTICS AND THE FORMAL ELABORATION BY BREWKHA

In 1977 Rescher’s book *Dialectics* appeared. Nowadays his book is highly praised in the field of computational dialectics, for the first time and most explicitly by Loui [1992]:

“By far, the most relevant work is the monograph of Nicholas Rescher (...) The formal system begs many questions. But is undoubtedly the most elegant system to date (...) I have found no further development of Rescher (...)”

Brewka [1994] took up the thread:

“We will therefore go back to Rescher and present a reconstruction of his theory. To do this we have to make precise what remained informal and vague in Rescher’s book.”

Both Rescher’s dialectics and Brewka’s formalization will be discussed.

A Rescher dispute takes place between two players, a proponent and an opponent, who take turns alternately. As in most dialog games, the dispute starts if the proponent claims a thesis. The roles of proponent and opponent do not change, so the starting player remains proponent throughout the game. The opponent’s role is to question the theses raised by the proponent. The role of the proponent is to take away the doubts of the opponent. If neither the proponent can put forward enough evidence for his thesis, nor the opponent can adduce decisive doubts, a determiner will have to decide who wins the dispute.

In the subsequent sections are discussed: the moves types, the dialog rules, the termination criteria, and the way commitment is dealt with.

The moves

There are three so-called fundamental moves.

<table>
<thead>
<tr>
<th>Type</th>
<th>Representation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Categorical assertion</td>
<td>P</td>
<td>P is the case</td>
</tr>
<tr>
<td>Cautious assertion</td>
<td>P</td>
<td>P is the case for all that you (adversary) have shown</td>
</tr>
<tr>
<td>Provisoed assertions</td>
<td>P/Q</td>
<td>P usually obtains provided that Q</td>
</tr>
</tbody>
</table>

The first move is exclusively for the proponent (indicated by ‘!’), the second move only for the opponent (indicated by ‘!’). Both players may perform the third move, that must always be accompanied with an assertion about the condition, so either:

P/Q & 1Q, or
P/Q & 1Q.

---

13 See for instance Vreeswijk [1993, p. 111]: “Here, we mainly draw upon Rescher’s excellent monograph (1977).”

14 Rescher does not use the term dialog game, but his formal dialectics can be considered to be one, see also [Woods & Walton, 1982].
The idea of Rescher’s game is that the proponent asserts statements (!), and the opponent raises doubts (†). In Brewka’s version this distinction has disappeared. Brewka omits the symbols ‘!’ and ‘†’, and instead labels what is said in the moves as either belonging to the proponent or the opponent. The consequence is that the role of proponent and opponent become identical ones. The asymmetry of Rescher’s game has become symmetry (see section 4.4).

In Brewka’s game the moves of the players are operations on sets. These operations consist of either adding elements to, or deleting elements from a set (or both). There is a set of defaults D that cannot be challenged, and a set of facts W. The set W consists of three subsets: the common set W, the proponent’s set W_pro, and the opponent’s set W_opp. The players may perform operations on these sets by way of the following three elementary move types:

- addD(i), ‘i’ is a fact or default; facts are moved to the current player’s set (g=W), defaults are moved to the set of defaults (g=D);
- concedeD(i), ‘i’ can only be a fact; the fact is moved from the opponent’s set of facts to the common set of facts;
- removeD(i), ‘i’ can only be a fact; the fact is removed from the current player’s set of facts.

Each move by a player is an arbitrary finite sequence of one of these elementary moves. So any combination of move types is allowed. Note that since there is only one set of defaults, added defaults are as if they were automatically conceded. Once a default is added, it cannot be removed.

4.2 Dialog rules

The rules of Rescher describe what the possible reactions are after a move. The reactions are split up into reactions to fundamental moves, and reactions to complex moves.

Reactions to fundamental moves

The first two fundamental moves, categorical assertions (!P) or cautious ones (†P), can be reacted to in two different ways. First, the assertion can be denied. Second, a default with as its conclusion ¬P can be claimed. The following scheme shows this, for a proponent who claimed P.

<table>
<thead>
<tr>
<th>Move</th>
<th>Reaction</th>
</tr>
</thead>
</table>
| !P    | 1. †¬P;  
       | 2. ¬P/Q & †Q |

The possible reactions to ¬P, †P and †¬P are the same two, except for the different prefix symbols. For instance, in reaction to the move †¬P, the proponent can either state !P, or put forward a default, P/Q & †Q.

---

15 Instead of the set W Brewka uses two sets: one of contingent facts C, and one of background knowledge T. For dealing with defeasibility this distinction may be necessary, for the present purposes the use of just one set of facts W (containing both C and T) suffices.
There is only one reaction possible to a provisoed assertion. That is the claim of a more specific provisoed assertion.

\[
\begin{array}{c|c}
\text{Move} & \text{Reaction} \\
P/Q & -P(Q & R)
\end{array}
\]

Rescher also mentions the need for a rule to prevent useless repetitions of moves. A special rule, called the blockage rule, says that a particular move may only occur once during a dispute. So the second move in the scheme below is forbidden.

\[
\begin{array}{c|c|c}
\text{PRO} & -P \neq \lnot P & \text{CON} \\
\lnot P & \lnot \lnot P & \lnot \lnot P
\end{array}
\]

**Reactions to complex moves**

The complex moves are built using fundamental moves. Therefore I will not discuss all the reactions Rescher describes to complex moves, but confine the discussion to a few additional remarks necessary for a complete definition of the game.

First, a provisoed assertion has to be accompanied by a categorical/cautious assertion. A reaction to this move may either concern the provisoed assertion, or the categorical/cautious assertion. For instance, a reaction to P/Q & \lnot Q can be:

1. An attack of the provisoed assertion P/Q: \(-P(Q&R) & \lnot(Q&R)\);
2. An attack of the categorical assertion \lnot Q: a. \lnot \lnot Q;
   b. \lnot Q/S & \lnot S.

Second, P&Q can be denied in three different ways. With the obvious, though ambiguous, \(-P&Q\). After this reaction it is not clear whether a player means that P&Q are both false, or that just either P or Q is false. This ambiguity does not exist if a player reacts in one of the other two ways: \-P or \-Q.

Finally, the proponent may react only to the last move of the opponent. The doubts raised by the opponent have to be dealt with immediately. The opponent, on the other hand, may react to other moves than the last one. In case the opponent is satisfied with a defense of the proponent, he can jump back to a previously attacked move. Consider for example the following dispute.

\[
\begin{array}{c|c}
\text{PRO} & \text{CON} \\
P/Q & \lnot Q \\
Q/R & \lnot R \\
\lnot P & \sim P(Q&S) & \lnot(Q&S)
\end{array}
\]

The opponent first denied the condition of the default P/Q. After he is satisfied with the proponent's explanation, he claims a more specific default in the second move, so in his second move he reacts again to the first move of the opponent.

Contradiction is not explicitly forbidden. The following sequence is possible.

---

16 Note that although a provisoed assertion is a fundamental move, in a dispute it can only be put forward in combination with another fundamental move, either a categorical or a cautious assertion.
The proponent added in the third move a default with a condition opposite to the condition of the default from the first move. Since the model aims to represent academic disputes, maybe such a contradiction will not occur anyway.

Brewka defines possible moves using a default theory for each player. The default theory of a player ($\Delta_x$, where $x$ is either 'pro' or 'opp') contains the set of defaults ($D$), the union of the set of facts agreed upon ($W$) and the set of facts added by this player that are not yet conceded ($W'_x$):

$$\Delta_x = (D, W \cup W'_x).$$

After a move by the player 'x' his new default theory $\Delta_x$ must be consistent. There is a game record, the so-called state of the game ($D$, $W$, $W'_pro$, and $W'_opp$), that is updated after each move.

Brewka allows an empty move, meaning that a player does not add, remove, or concede. In that case it is the previous player's turn again. The purpose of the empty move is the same as what questions are for in DiaLaw.

Repetition of moves is forbidden: after a move the new state must be different from each previous state, except the last one (allowing empty moves, but not two subsequent empty moves). If this condition is not met, a move is illegal.

**Termination criteria**

The proponent wins a Rescher dispute if he successfully defended his initial assertion. This is the case if the opponent cannot perform an attacking move anymore. The opponent wins if he successfully attacked the initial assertion. This is the case if the proponent is not capable to defend anymore. If one of these situations occurs, a determiner decides who is the winner. The role of the determiner is not really worked out but comes down to the following. He decides on formal grounds (e.g., illegal moves, like arguing in circles) or on material grounds (e.g., the extent to which the opponent forced the proponent to implausible commitments).

Brewka uses the default theories of the players to define the end of a dispute. The proponent wins a dispute about a formula if after a move the formula is a skeptical consequence\(^\text{17}\) of the opponent's default theory. Conversely, the opponent wins if after a move the formula is not a skeptical consequence of the proponent's default theory.

### 4.3 Commitment

Rescher uses commitment in a comprehensible, but uncommon way. Namely, to indicate what the proponent has to support at a certain moment: "... theses to whose defense he is committed." A common consequence for a player who claims is that he becomes committed. The same holds in Rescher's system. However, the commitment changes during

\(^{17}\) A formula is a skeptical consequence if it exists in all extensions of a default theory.
Dialogical models of argumentation

the dispute in a rather unusual way. If the proponent supports P by P/Q & !Q, the commitment to P expires and is shifted to Q. In most systems the proponent would now be committed to both P and Q.

Because commitment is defined in terms of defense, the players are not committed to the indisputable provisoed assertions like P/Q. Normally, in case of indisputable assertions, both parties would be committed.

Reischer distinguishes three types of commitment:

• proponent’s undischarged commitments;
• opponent’s concessions;
• proponent’s living commitments.

The undischarged commitments of the proponent are the assertions the opponent may challenge. So after !P, an undischarged commitment of the proponent is P. As was just described, after P/Q & !Q the proponent has no longer the burden of proving P, but instead is committed to Q. The reason he has no longer the duty to defend P, is that it is supported by Q and the indisputable P/Q.

The opponent’s concessions are the assertions he did not explicitly attack.

<table>
<thead>
<tr>
<th>PRO</th>
<th>CON</th>
</tr>
</thead>
<tbody>
<tr>
<td>punish/killer &amp; !killer</td>
<td>a. ~punish(killer &amp; king) &amp; !killer &amp; king</td>
</tr>
<tr>
<td></td>
<td>b. ↑~killer</td>
</tr>
</tbody>
</table>

In this example the opponent is after the first reaction (a) committed to killer, because he did not attack killer. Instead he attacked punish/killer by claiming a more specific default, namely that in case the killer is a king he cannot be punished. After the second reaction (b) he is no longer committed to killer, because he attacked killer by claiming ~killer. Finally, the proponent’s living commitments are a subset of his undischarged commitments. If the opponent has attacked an assertion, the undischarged commitment is also a living one. For instance, after the second reaction (b), the statement killer is a living commitment.

Reischer’s restricted scope of commitment is defensible. Normally commitment is used to force opponents, or, as in MacKenzie’s system [1979a], to confront the opponent with inconsistencies. In Reischer’s system possible attack is limited, and the dispute is very structured. Players can only attack theses that are not supported by a provisoed assertion, or attack provisoed assertions by putting forward more specific ones. That is why commitment of the proponent is only necessary concerning the assertions he has not yet supported.

Although Brewka does not use the notion commitment, he uses four sets for the same purpose as normally commitment stores are used. First, one with the defaults both players are committed to. Then, three sets with facts: one for each player, and a common set. The two sets containing commitment of either player are similar to MacKenzie’s commitment stores. Instead of having shared commitment stored in both individual sets, Brewka uses common sets. The advantage of these common sets is that they clearly show what the players agree upon. Brewka’s commitment is different from Reischer’s, and more like DiaLaw: players are committed to what they claim and concede, and lose commitment after withdrawal.

18 Instead of these three sets (W, W_pro, W_opp), Brewka uses six sets (C, C_pro, C_opp, T, T_pro, T_opp). See supra, note 15.
4.4 Concluding remarks

Rescher tried to capture the process of (academic) disputation in formal dialectics. The proponent makes claims all the time, and the opponent only has to be skeptical about what has been put forward. In Brewka’s system this asymmetry is gone, because both players have the same role.

Brewka’s contribution is that his formal approach makes it possible to indicate exactly when a dispute is won or lost, and when a move is allowed or not. The way he uses default theories to do so is similar to Gordon’s approach (see section 5). Some changes proposed by Brewka, for instance leaving out ! and ⊤, introducing the empty move and explicit consent, makes his system different from Rescher’s on essential points.

Brewka turned the disputes into a symmetric dialog game. According to Rescher a symmetric game is a combination of two asymmetric games, and therefore cannot offer any new insights. I think on this point he underestimates the power of dialog models in which both players can take the initiative. Consider, for instance, the following short dialog.

<table>
<thead>
<tr>
<th>PRO</th>
<th>CON</th>
</tr>
</thead>
<tbody>
<tr>
<td>P!</td>
<td>¬P/Q &amp; ⊤Q</td>
</tr>
</tbody>
</table>

In case the proponent knows nothing about Q, and he wants to know more about Q, he can only get this information if he reacts with P. But that would mean that he claims that ¬Q is the case, while he just does not know anything about Q. It would be better here if he could question Q, so assert ⊤¬Q. But since the game is asymmetric he cannot do this.

An improvement that Brewka suggests allows the following dispute.\(^\text{19}\)

<table>
<thead>
<tr>
<th>PRO</th>
<th>CON</th>
</tr>
</thead>
<tbody>
<tr>
<td>P/Q &amp; ⊤Q</td>
<td>¬P/R &amp; ⊤R</td>
</tr>
</tbody>
</table>

The dispute ends undecided after these two moves, but according to Brewka the issues are now clear. It would be a real improvement if the dialog could continue after determining that there are contradicting defaults, like is possible in DiaLaw.\(^\text{20}\)

Defaults are indisputable in Rescher’s game (and Brewka’s formalization). That defaults cannot be attacked is not because Rescher thought it unnecessary to argue about the truth of defaults, but because he wanted to concentrate on the core of a dispute. If he would have included disputable defaults, the structure of the disputes would have been less clear than it is now.

\(^{19}\) The notation of Rescher is used.

\(^{20}\) The example Brewka uses is known as the Nixon paradox: what to decide in case Democrats are no pacifists, Quakers are, and Nixon is both a democrat and a quaker? In DiaLaw the dialog could continue at the point Brewka stops, namely after determining: reason(quaker, pacifist), reason(democrat, ¬pacifist). In DiaLaw the discussion can, for instance, continue in this way: outweigh(democrat, quaker, ¬pacifist). So a player can claim that the democrat argument outweighs the quaker argument, or the other way around. The dialog may proceed with discussing why this outweigh relation is justified.
5. Gordon's Pleadings Game

In 1993, when the field of Artificial Intelligence and Law became interested in the representation of law in dialogical or dialectical models (see, e.g., Hage, Span & Lodder, 1992; Loui, 1992; Nitta, Wong & Ohtake, 1993), Gordon completed his dissertation in which a formalization and implementation of a dialogical model of law is described: the Pleadings Game.

Prakken [1995] suggests that models like Gordon's are inspired by an advice of Toulmin.21 Gordon's close colleague Brewka [1994], however, has another opinion about the origin of the Pleadings Game: "The roots of Gordon's approach can be traced back to the philosopher Nicholas Rescher."22 The following quotes from Gordon himself leave no doubt about the roots and inspiration:

"The Pleadings Game takes another approach, founded in the discourse theory of legal argumentation developed by Robert Alexy" [Gordon 1995, p. x]

"(...) our goal is a normative model of pleading, founded on first principles, inspired by Robert Alexy's discourse theory of legal argumentation." [Gordon 1995, p. 112]

The Pleadings Game is reported on in several publications. A clear, short introduction is his ICAI contribution [Gordon, 1993a]. His dissertation [Gordon, 1993b] offers a full discussion and is -with some minor changes- republished in [Gordon, 1995]. In his dissertation Gordon defines the Pleadings Game, after evaluating theories of legal philosophy and formal systems of argumentation. Here I discuss the definition of the Pleadings Game23 following Gordon [1995], for it is the best obtainable source.

One of Gordon's objectives was to refine Alexy's [1989] theory of legal argumentation to make it serve as a model for computer implementation. A difference with Alexy's general theory is that Gordon restricted his attention to a small part of the legal discipline: civil pleading.

Civil pleading can be characterized as the attempts of two parties in a civil suit to convince their opponent. The legal domain of Gordon's pleadings is Article Nine of the Uniform Commercial Code (UCC) of the United States, that covers secured transactions. The Pleadings Game mediates the process of exchanging arguments between the two players (plaintiff and defendant).

The purpose of the Pleadings Game is to identify on what points the parties do not agree, or, in legal terms, determine what issues -both legal and factual- exist between the parties.

---

21 Toulmin's [1958, pp. 7-8] advise is that logicians who want to learn about reasoning in practice, should turn away from mathematics and instead study jurisprudence. However, Gordon is not a logician who needed an appropriate domain for his logic, but a lawyer who used logic to study law.

22 Gordon [1995, p. 107] mentions Rescher only once, "Rescher's system was an early attempt to handle defeasible arguments and the first to rank them by specificity."

23 See also [Gordon, 1994].
5.1 An informal description of the Pleadings Game

The Pleadings Game starts after the plaintiff files his complaint. In this complaint, referred to as the main claim, the plaintiff determines the subject of the dialog. The first move in the dialog is the defendant’s reaction to the main claim. The defendant can either concede or deny the main claim. The Pleadings Game ends if the defendant concedes the main claim, but most often he will deny. Denying does not mean that the player claims that the opposite statement is true, it only means that he is not convinced (yet). In other words, he wants further evidence for the statement, that is he wants his opponent to justify the statement.

After the defendant denied the main claim, the plaintiff adduces an argument supporting his complaint. An argument is a set with the following content:

1. a formula saying that some particular rule is valid;
2. formulas corresponding to the condition of this rule;
3. a formula saying that this rule is applicable.

The application of a valid rule creates an argument supporting the conclusion of the rule.

If the plaintiff has made his argument, it is the defendant’s turn once more. In reaction to the argument of the plaintiff the defendant has to perform the following assertions in his next turn.

1. He must either deny or concede that the rule is valid. If the validity of the rule is conceded, the validity becomes indisputable. If the validity of the rule is denied, in the next move the plaintiff will have the opportunity to adduce an argument supporting the claim that the rule is valid.
2. He must concede or deny the condition(s) of the rule. The consequences of these assertions are the same as above (1).
3. He must concede the argument, or adduce a counterargument. Conceding has the same consequence as above (1). Note, however, that if an argument is conceded, it does not necessarily imply that the conclusion of the argument follows. This is only the case if there is also agreement about both the validity of the rule and the satisfaction of the conditions of the rule. The other possibility is to adduce a counterargument. The counterargument is composed of exactly the same elements as the argument described above. The conclusion of the counterargument is opposite to the conclusion of the argument.

The process of both parties conceding and denying statements, and exchanging arguments (counterarguments, countercounterarguments, etc.) continues until each statement is:

- a) conceded
- b) twice denied
- c) countered

If a statement is conceded, there is agreement about the statement. In that case there is no need to discuss this statement anymore.

---

24 The plaintiff can also deny the denial. In that case the Pleadings Game ends with only one issue: the main claim. Surely, the plaintiff will know arguments that support his claim, because otherwise he would not have started the Game.
If a party denies a statement, and the other party denies this denial, there is also no need to proceed any further. Otherwise, meaningless yes/no-'discussions' would be the result. So, if a statement is twice denied, disagreement is established. In the Pleadings Game this means that a decision about the statement is left open for trial.

Finally, a statement can be countered. At present it suffices to say that a denial is countered by an argument supporting the statement that was denied by the opponent, and that an argument can be countered by a counterargument. When discussing the moves, the possible counters are discussed in more detail.

If each statement is reacted to in the above a-c described way, the Pleadings Game ends, because then there are no statements left open for discussion. Consequently, a player's turn is the last of the game if he has introduced no new statements. This is the case if:
1. no argument is adduced;\(^{25}\)
2. no statement is denied, except denials.\(^{26}\)

5.2 The moves

In the Pleadings Game moves are assertions about statements. There are four different types of moves.\(^{27}\) First, if a player thinks that a statement of his opponent is true or acceptable, he concedes it.

Second, if he is not sure about a statement of his opponent, he can deny it. In the informal description the meaning of deny was already explained. Denying is comparable to Rescher's cautious assertion, and to question in DiaLaw.

Third, a player can declare a rule. If a player declares a rule, he claims that the way he represents the rule is accurate. A rule expresses a general relation between a condition and a conclusion. A rule is not necessarily based on legislation, but can be based on a legal principle as well, or on whatever other general relation the player likes to express. Rules play an important role in the Pleadings Game, because arguments are based on rules. The relation between a rule and an argument is comparable to the relation between a rule and a reason in Reason-based Logic. This brings us to the move in which arguments are introduced.

An argument is introduced as a request to the opponent to defend his statement against the argument. So, an argument only indirectly supports one's own standpoint, and directly attacks the statement of the opponent.\(^{28}\) When the dialog rules are discussed, examples of how arguments are put forward are given. The four possible moves are presented in the following scheme, which includes their formal representation.

---

\(^{25}\) Each introduced argument brings with it at least a new statement that expresses the argument.

\(^{26}\) If a statement is denied, a new statement, viz. the denial, becomes part of the game. That a denial may be denied follows from what is said about statements which are denied twice (see b).

\(^{27}\) Gordon calls the moves assertions, but for the sake of uniformity I refer to these assertions as moves.

\(^{28}\) An argument asserted as defend((denial (claim p)) A) is stored as (argument A p).
Whether a move is allowed is formulated in the dialog rules, as well as what the consequence of a move is. Because the moves are about statements, and the term statement is used throughout my discussion of the Pleadings Game, I show the different statements in a scheme similar to the one used for moves.

<table>
<thead>
<tr>
<th>Formal</th>
<th>Meaning</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(concede s)</td>
<td>s is accepted</td>
<td>s must be a statement</td>
</tr>
<tr>
<td>(deny s)</td>
<td>challenges that s is the case</td>
<td>see 1</td>
</tr>
<tr>
<td>(defend s A)</td>
<td>defend s with A</td>
<td>see 1, and A is a set of formulas (of the language L)</td>
</tr>
<tr>
<td>(declare r)</td>
<td>r is a valid rule</td>
<td>r must be a rule</td>
</tr>
</tbody>
</table>

5.3 Dialog rules

5.3.1 Preliminary remarks

The Pleadings Game does not have a record in which the played moves are stored. Instead, the consequences of the moves are stored in what is called the game record. This record is a triple \( <b, \pi, \delta> \), where \( b \) is the background of the game, and \( \pi, \delta \) are the statements of the plaintiff and the defendant, respectively.\(^{29}\)

The background \( b \) is a triple \( <\phi, S, R> \), where \( \phi \) is the main claim of the game (the claim filed by the plaintiff at the beginning of the Pleadings Game), \( S \) is the set of formulas and \( R \) the set of rules the players agree upon. Both sets may be empty, so the initial background can be \( <\phi, \emptyset, \emptyset> \). However, the background knowledge can also be filled with formulas and rules agreed upon in a pre-Pleadings Game negotiation, which is “some unspecified procedure” [Gordon 1995, p. 141].

The statements of a party are stored in the following triple: \( <O, D, C> \). For each party there is a triple in which the statements he introduced in the game are stored. The set \( O \) contains the open statements: the other player has not yet answered them. This set is similar to the set of living commitments in Rescher’s game, the set of disputed sentences in

\(^{29}\) Because it is not allowed to retract statements, I think that at each stage of the game it is possible to reconstruct exactly what the previous moves were
Dialogical models of argumentation

DiaLaw, and the sets that are attached to only one player in Brewka’s game. The sets $D$ and $C$ contain statements that are denied and conceded by the other party, respectively.

The player who’s turn it is, no matter whether it is the plaintiff or the defendant, is called the proponent of the move, the other player the opponent. The role of proponent and opponent therefore changes after each turn. A single turn can contain more moves.

53.2 Conditional entailment

The notion of arguments in the Pleadings Game is defined using Geffner & Pearl’s [1992] conditional entailment. My discussion of conditional entailment is not complete, but includes what is necessary for understanding the Pleadings Game.

A default theory $T$ is a tuple $<K, E>$, where $K$ is the background (containing rules), and $E$ is the evidence (containing facts). The background $K$ is also a tuple $<L, D>$, where $L$ is a set of strict rules and $D$ is a set of default rules.

Default rules are represented in conditional entailment as follows:

1. sun_shines $\land d_1 \rightarrow$ nice_weather;
2. sun_shines $\Rightarrow d_1$.

The first rule is a material implication, so an element of the set of strict rules $L$. The second rule is defeasible, so an element of the set of default rules $D$. The two rules mean: 1) if the sun shines and the default $d_1$ applies, then it is nice weather; 2) if the sun shines, then normally the default $d_1$ applies.

Rules may conflict. For instance, the rule “if it rains, then normally the weather is not nice” has a conclusion opposite to the previous rule. This rain rule can be represented as:

1. raining $\land d_2 \rightarrow \neg$ nice_weather;
2. raining $\Rightarrow d_2$.

Suppose that the facts $(E)$ are $(raining, sun\_shines)$. In that case default 1 justifies the conclusion that it is nice weather, while default 2 justifies the conclusion that it is not nice weather. In that case there are two models. One in which the conclusion that it is nice weather is justified, and one in which the conclusion that it is not nice weather is justified. If there is no more information, it cannot be decided which model is to be preferred: the model with nice_weather or the model with $\neg$ nice_weather.

In general, the model is preferred in which the most defaults are applied, or, in other words, the least defaults are excluded. In our example one default is excluded in each model. But, if there had been more defaults, and in one model only one default would have been excluded, while all the other models would exclude at least two defaults, the former model would be preferred. There can be an ordering on models.

In conditional entailment ordering on models is handled in the following way. Defaults can be explicitly prioritized. If in the example the default $d_2$ would have been preferred, this is expressed as that the default $d_2$ has a lower priority: $d_2 < d_1$. This means that if both defaults are applicable, those models in which the default $d_1$ is applied, are preferred to those in which the default $d_2$ is applied.

30 Conditional entailment combines the benefits of conditional interpretation (allows ranking of arguments) and extensional interpretation (ignores irrelevant information).
5.3.3 Arguments in the Pleadings Game

The rules the players declare in the Pleadings Game have a special format, in order to make it possible to use conditional entailment. Each rule is split into a default part (element of the set of defaults $D$), and into a strict part (element of the set of non-defeasible, strict rules $L$).

The rule that someone who kills on purpose is a murderer, unless he acted in self-defense, is declared in the Pleadings Game as follows:31

\[
\text{declare (rule sr-289-41 (x y))}
\text{if (killed x y)}
\text{then (murderer x)}
\text{unless (self-defense x))}, \tag{32}
\]

The rule has five parts: the name of the rule, its free variables, the condition of the rule, the conclusion of the rule, and –optional– an exception. If the declaring player knows that the exception does not apply, he can leave it out.33 However, a problem with leaving out the exception is that the other player may argue that the rule is not represented correctly.

In terms of the set of $L$ and $D$ this rule is translated to:

\[
L = \{ (\text{all (x y)}
(\text{if (and (killer x y) (backing sr-289-41)})
(\text{antecedent (inst sr-289-41 (parms x y)))})}, \tag{1}
\]

\[
(\text{all (x y)}
(\text{if (and (killer x y) (backing sr-289-41)}
(\text{ap (inst sr-289-41 (parms x y)))})
(\text{applies (inst sr-289-41 (parms x)))}), \tag{2}
\]

\[
(\text{all (x)}
(\text{if (applies (inst sr-289-41 (parms x))})
(\text{murderer x)}), \tag{3}
\]

\[
(\text{all (x)}
(\text{if (and (self-defense x)}
(\text{ap (inst sr-289-41 (parms x)))})
\text{false}))
\}
\]

\[
D = \{ (\text{antecedent (inst sr-289-41 (parms x))) \Rightarrow}
(\text{ap (inst sr-289-41 (parms x)))}
\}
\]

The meaning of the four strict rules ($L$) is:

---

31 Because this rule is based on a combination of two rules from the Dutch penal code, art. 289 Sr and art. 41 Sr, it is referred to as sr-289-41.

32 The representation of the rule is according to the convenient denotation [Gordon, 1995, p. 117].

33 (rule d if a then c) instead of (rule d if a then c unless e)
1. If someone is a killer and the rule is backed (derived from Toulmin's [1958] backings), then the condition of the rule is satisfied;
2. If someone is a killer, the rule is backed, and the rule is applicable, then the rule applies;
3. If the rule applies, then someone is murderer;
4. If the rule is applicable and someone acted in self-defense, the conclusion cannot be derived (false).

The default should be read as: if a rule is backed and the conditions of the rule are satisfied, then normally the rule is applicable.

One of the consequences of the declaration of a rule is that the backing of the rule is added to the set of open statements of the opponent, viz.:

(claim (backing rule sr-289-41)).

The opponent is forced to accept each rule declared by the proponent. So, all that is in L and D above becomes part of the set of conceded statements. However, if a player is not satisfied with the representation of the rule, he can deny the backing of the rule. Note that although a player is committed to a declared rule, the rule can only have effect if both the backing and the conditions of the rule are agreed upon.

The following example illustrates how an argument is put forward in the game. Suppose the (main) claim is that Albert is a murderer, and that the defendant has denied the claim. If the plaintiff\textsuperscript{35} puts forward the argument that Albert killed his mother (and therefore is a murderer) this is represented as:

\begin{verbatim}
(defend (denial (claim (murderer albert)))
  | (killed albert mother)
    (backing rule sr-289-41)
    (ap (inst sr-289-41 (parms albert mother))))
)
\end{verbatim}

This assertion reads as: "defend the statement denying Albert is a murderer, against the application of the valid rule sr-289-41". A consequence of this move is that the statement:

(\begin{verbatim}
(argument ((killed albert mother)
  (backing rule sr-289-41)
  (ap (inst sr-289-41 (parms albert mother)))
  (murderer albert)),
\end{verbatim}

becomes part of the set of open statements of the plaintiff.

It is not allowed to deny an argument. Denying means asking for justification, and in the Pleadings Game only justified or real arguments and rebuttals are admitted. An argument is justified if the conclusion of the argument is known to be entailed by the argument, given

\textsuperscript{34} In the example the claim is represented as \textit{(claim (backing sr-289-41))}. However, because the general form is \textit{(claim (backing r))}, the backing actually has to be \textit{(claim (backing rule sr-289-41 \(x\ y\) if \(killed x\ y\) then \(murderer x\)) unless (self-defense x))}.

\textsuperscript{35} I stick to the terminology of the Pleadings Game. Because this is a penal case, the plaintiff should actually be called the prosecutor, and the defendant the accused.
the current background context. Known is a formally defined weak entailment relation. A formula \( p \) is known to be entailed by an argument \( A \) if:

1. \( p \) is an element of the union of \( A \) and the background context, or;
2. it is known that the argument \( A \) causes an inconsistency, or;
3. there is an argument \( B \) that supports \( p \), and each formula of \( B \) is known to be entailed by \( A \).

In the detailed example given by Gordon only the first known relation is used. Here also an example of the first known relation is given. Let us determine whether the argument supporting the statement that Albert is a murderer was justified. The statement that Albert is a murderer must be an element of the union of the argument \( A \),

\[
{(\text{killed albert mother}),}
{(\text{backing rule sr-289-41}),}
{(\text{ap (inst sr-289-41 (parms albert mother))})}
\]

and the background context (see section 5.3.1): the formulas in \( S \), and the rules in \( R \). The set of rules \( R \) contains the four strict rules of \( L \) and the defeasible rule of \( D \) (see above).

The condition of the rule sr-289-41 is satisfied and the rule is backed. The default that the rule is applicable is not defeated by any other default (there are no other defaults yet), so the rule is applicable. Because the rule is applicable, the rule applies. Finally, because the rule applies and the exception does not occur, the conclusion, \( \text{murderer albert} \) can be derived. So \( \text{murderer albert} \) is an element of the union of the argument and the background context. Because the conclusion \( \text{murderer albert} \) is known to be entailed by the argument, the argument really is an argument.

All formulas of an argument, except the applicability assumption,\(^{36}\) are claims in the set of open statements of the proponent. If the plaintiff adduced the just mentioned argument about Albert who killed his mother, a consequence is that,

\[
(\text{claim (killed albert mother)})
(\text{claim (backing rule sr-289-41)}),
\]

become elements of the open set of statements of the plaintiff.

A rebuttal (counterargument) is defined the following way: the union of the argument and the rebuttal must be false. If an argument is rebutted by a rebuttal \( R \), in the next move the rebuttal can be countered by an argument that attacks the rebuttal (and supports the rebutted argument). Next, this new argument can be rebutted, etc.

Arguments are checked before they are introduced in the game. Only real arguments/rebuttals are allowed. A rebuttal may appear to be 'false', because the contradiction is caused by the rebuttal solely (and not because of the union of the rebuttal

---

\(^{36}\) The applicability assumption is an assumption that, in the face of new information, may be defeated. For instance, a rule is no longer applicable if an exception occurs. Since the retraction of claims is not possible, commitment to the applicability assumption would lead to an inconsistency. In order to prevent that inconsistencies arise, the applicability assumption does not become part of the set of open statements.
and the argument).\textsuperscript{37} Nevertheless, also in this case the rebuttal is a real rebuttal in terms of Conditional Entailment.

5.4 Relevancy and issues

During the Pleadings Game the players are not allowed to assert everything they like. The proponent of the move can perform one or more of the following moves:

1. forced - react to relevant open statements of the opponent;
2. optional - react to open statements of the opponent that are not relevant;
3. optional - declare (an arbitrary number of) rules.

A statement is relevant if it is about an issue. For the four different statements, \((\text{claim } p), (\text{denial}(\text{claim } p)), (\text{argument } A \ p)\) and \((\text{rebuttal } A \ p \ C)\), this means that they are relevant if \(p\) is an issue.

In order to explain how issues arise and expire, I introduce the notion of a decisive argument or rebuttal. An argument (rebuttal) is decisive, if the opponent:

- conceded the argument (rebuttal), and;
- either conceded the backing of the rule, or is committed to this backing because of a decisive argument supporting it, and;
- either conceded the conditions of the underlying rule, or is committed to the conditions of the underlying rule because of a decisive argument supporting it.

Note that since each statement may be reacted to only once, it is impossible that a conceded argument is also countered. So in case of a decisive argument, which is an argument that is conceded, there can be no counterargument.

\[
\begin{align*}
\text{(claim } p) & \quad \text{p is an issue} \\
\text{deny} & \quad \text{concede} \\
\text{decisive argument} & \quad \text{no decisive argument} \\
\text{p is justified} & \quad \text{p remains issue} \\
\text{decisive rebuttal} & \quad \text{no decisive rebuttal} \\
\text{p is not justified} & \quad \text{p remains issue}
\end{align*}
\]

If the opponent concede a claim, the formula that the claim is about is no longer an issue; the formula \(p\) is justified.

\textsuperscript{37} If the rebuttal is itself contradicting, it has to be defended against an empty argument. One of the consequences of adding a false rebuttal to the game, is that (rebuttal R false) becomes part of the set of conceded statements.
After a denial there are three different scenarios. First, if the denial is denied, the formula remains an issue that has to be decided in the Trial Game.

Second, if a decisive argument has been put forward, the formula \( p \) is justified. A decisive argument is not necessarily the argument put forward in reaction to the denial, but can also be adduced after one or more rebuttals. An argument has the following structure: \((\text{argument}(A_n \cup R_{n-1} p))\), where \( n \geq 1 \). If \( n = 1 \) the first argument was successful, and there was no rebuttal; if \( n = 2 \) then the first rebuttal is successfully countered. In general, if \( n = k \) then the \((k-1)\) rebuttal is successfully countered.

Third, it may be the case that the argument is not decisive. What happens then to the formula \( p \), depends on the opponent’s reaction. If he has put forward a decisive rebuttal, it is established that the formula \( p \) cannot be justified. A rebuttal has the following structure: \((\text{rebuttal } A_n \ p R_n)\), where \( n \geq 1 \). If \( n = 1 \) the first rebuttal was successful, if \( n = 2 \) the second rebuttal was successful. In general, the \( n \)-th argument is successfully rebutted. In case the argument was not rebutted, or no decisive rebuttal has been put forward, the formula \( p \) remains an issue.

5.5 Commitment

Gordon’s way of handling commitment is rather complicated. Each player is committed to statements in four different sets. Recall that the statements of a player are stored in the triple \( \langle O, D, C \rangle \).

The proponent (\( p \)) is in the first place committed to all statements he claimed and the opponent did not respond to (\( O_\circ \)), deny (\( D_\circ \)), or concede (\( C_\circ \)). Furthermore, he is committed to statements claimed by his opponent (\( o \)) he did concede: \( C_o \). The commitment of the opponent (\( o \)) is similar, and also stored in four different sets: \( O_o, D_o, C_o, \) and \( C_p \).

In addition to the set of statements belonging to each player, there is also the background of the game \( \langle \Phi, S, R \rangle \) (see section 5.3.1), where \( S \) is the set of formulas and \( R \) the set of rules the players agreed upon. If a statement is an element of either \( C_o \) or \( C_p \), the corresponding formula is an element of \( S \). For instance, if the statement (claim \( p \)) is in \( C_o \) then the formula \( p \) is an element of \( S \). Since a declared rule is conceded automatically, all declared rules are an element of \( R \).

The players are also committed to the logical consequences of their claims. This consequence relation is not traditional entailment, but the weaker previously described-known relation. In fact, Gordon [1995, p. 110, p. 204] claims that one of the novel features of the Pleadings Game is that “inference is used to commit players to the known consequences of their claims”. This strengthening of commitment takes place if a statement in the open set of the opponent is known to be entailed by the statements of the proponent. In that case the proponent becomes committed to that statement too. Furthermore, a claim does not become an element of the set of open statements of the proponent if it is already known to be entailed by his own claims. Although theoretically of interest, unfortunately in Gordon’s detailed example strengthening of commitment does not occur.

5.6 Concluding remarks

The Pleadings Game is undeniably a major achievement. Not only the possibility and the need to represent legal reasoning dialogically are discussed, also a formalization and
implementation are described. In this section concluding remarks are made about different aspects of the Pleadings Game.

**Deny and defend**

The meaning of deny is somewhat ambiguous. It is not meant as a negation, like \( \text{claim } \neg p \) would be as a reaction to \( \text{claim } p \). Rather, it is a request for the opponent to bear his burden of proof for some formula. Conversely, the denied statement seems to be a sort of negation as well. Namely, the player who denied a claim may not use it in his own arguments. This repercussion seems to be too strong for a player who only expressed not to be convinced at a certain point in the game.

I think that the assertions would become more natural if denial would be replaced by a move question that does not change the content of the sets of statements.

<table>
<thead>
<tr>
<th>Pleadings Game</th>
<th>Suggestion</th>
</tr>
</thead>
<tbody>
<tr>
<td>(deny(claim p))</td>
<td>(question(claim p))</td>
</tr>
</tbody>
</table>

Another strange move is the defense move. An argument \( A \) that supports the formula \( p \), stored as the statement \( \text{argument } A \ p \), is introduced in a somewhat unnatural way.

\[
\text{defend} \ (\text{denial(claim p)}) \ A
\]

If deny is replaced by question, arguments can be put forward in a way that is more isomorphic to how they are stored. For instance, instead of the above defend move, the argument could be put forward as:

\[
\text{assert(argument } A \ p \).
\]

The conditions and consequences of all moves can remain largely unaltered. The replacement of defend with assert leads to the following changes.

<table>
<thead>
<tr>
<th>Pleadings Game</th>
<th>Suggestion</th>
</tr>
</thead>
<tbody>
<tr>
<td>(defend(denial(claim p)) A)</td>
<td>(assert(claim A p))</td>
</tr>
<tr>
<td>(defend(claim A p) ( R_n ))</td>
<td>(assert(rebuttal ( A, R_n, B ))</td>
</tr>
<tr>
<td>(defend(rebuttal ( A, p ) ( R_n )) {} }</td>
<td>(assert(claim ( R_n ) false))</td>
</tr>
<tr>
<td>(defend(rebuttal ( A, p ) ( R_n )) }</td>
<td>(assert(claim ( A_{n+1} p )) }</td>
</tr>
</tbody>
</table>

**Possible measures to decrease necessary moves**

In civil pleading a party that does not react to a statement becomes committed. Therefore it would have been more in accordance with practice if the silent implies consent principle had been used. An additional advantage would be that in that case a player has to perform significantly fewer moves.

It could be claimed that the number of statements a player must react to is already diminished by the use of the notion of relevancy. A player only has to react to all relevant open statements. However, given the constraints of the game, it seems that irrelevant open statements will hardly occur. For instance, in the detailed example described by Gordon at no stage of the game there are irrelevant open statements.

If an argument is added, all formulas of the argument (except the applicability assumption) become claims that must either be conceded or denied. If a party knows one of them to be false, he has no opportunity to prove this immediately. Instead, he must wait for the proof of his opponent. It would speed up the game if it would be allowed to adduce arguments against claims. It is rather artificial that a player who has evidence that a

\[38\] Note that \( A_{n+1} = A_n \cup R_n \cup B \)
statement is not true, may not put forward his evidence immediately, but first has to ask his opponent to bear his burden to prove that the statement is true.

**Representation of rules**

In law it is important that a discussion about the validity of rules is allowed. The Pleadings Game offers this opportunity. Gordon realizes this by adding as a condition to a rule that it is backed. Moreover, not only the backing of the rule, but the whole procedure of rule application is part of the conditions of the rule. I think that the condition of a rule contains too much information that should not be part of the condition. Instead, rules could be modeled as in DiaLaw: the validity or applicability of a rule is not part of the condition of the reason, but is a different statement that can be used to justify the reason.

The way exceptions are modeled is inspired by Toulmin. The disadvantage of this representation is that exceptions occur in two different ways. First, the exception can be explicitly represented (unless) in the rule. Secondly, the exception can be modeled as a rule.

\[
\begin{align*}
&\text{if } a \text{ then } b \text{ unless } e \\
&\text{if } e \text{ then not } b
\end{align*}
\]

I think that a player will not easily represent a rule with an explicit exception. If he does, he gives away one way the argument can be attacked. He can better leave the exception out and wait for his opponent to come up with the exception in a different rule.

The drawback of allowing only one reaction is that the outcome of a debate may be opposite to the outcome if more arguments were allowed. For instance, a player who by mistake produced an argument that is defeated loses the conflict, although he might have more strings to his bow. An advantage of the constraint of only one reaction, is that the Pleadings Game may end sooner. Without losing the debate a non-constructive player that only denies can be stopped, simply by denying his denials. This means that infinite regress can be stopped without admitting that the statement one tries to justify is not true. This is really a strong point.

A last remark concerns the players of the Pleadings Game. Because of the difficulty of the underlying logic (Conditional Entailment), and the great number of statements a player has to react to in one turn, it takes almost super humans to play the game. It certainly takes time to learn the game. However, the ideas behind the Pleadings Game are valuable. And once one knows how to play the Pleadings Game, the game itself is valuable too. Because in playing the game one can learn a lot about legal reasoning.\(^\text{39}\)

6. **Survey of moves and commitment**

A survey of moves and commitment of the games discussed in section 3-5 and of DiaLaw is given, including tables showing the properties of the games. First the moves are discussed, second the commitment.

---

\(^{39}\) According to Gordon [1996] the game is not intended to be really played. It has theoretical purposes.
6.1 Moves

In DiaLaw there are four move-types: claim, accept, withdraw and question. Because these types are rather basic, the moves of the other games just described (section 3-5) are compared to DiaLaw on these four points: claims, concessions, retractions and questions.

Claims
The move assert or claim provides the food for the discussion. It is the move by which new information is added to the game. Since this move is essential, each game contains a move of this type.

I will make a distinction between the claim of statements, the claim of rules, and the claim of arguments. The reason to treat them differently is in the first place because some games have a different move for the introduction of rules (Gordon, Rescher). Moreover, in other games rules have a special status because they are above dispute (Brewka). Gordon also has a special rule for the introduction of arguments.

In MacKenzie's game statements are introduced by the move Statement. Rescher introduces statements in the assertion moves: categorical if made by the proponent, cautious if made by the opponent. Where Brewka has the move add, in DiaLaw the move claim is used.

In the Pleadings Game statements are only introduced indirectly; the consequence of a move can be that a statement is added.

The material implication in MacKenzie's game can be considered as a rule. The material implication is introduced with the move statement, but can also be introduced indirectly. 40 Rescher has a special move for the introduction of defaults: provisoed assertions. They are always accompanied with an assertion stating that the condition of the default holds. Finally, in the Pleadings Game rules are declared.

DiaLaw has no special move for the introduction of rules (claim), neither has Brewka for the introduction of defaults (add).

The Pleadings Game has a special move, defend, in which arguments are introduced. In DiaLaw the reason and outweighs predicate are introduced by the normal claim.

Concessions

In DiaLaw justification is based on agreement. Since most players are convinced of being right, they primarily want their opponent to agree with them. DiaLaw knows the move accept, Brewka and Gordon the similar move concede.

Because the principle that silent implies consent is an element of MacKenzie's game, there is no move for concessions. A similar technique is used in Rescher's game, that also lacks an explicit accept move. If a statement is not attacked, it is supposed to be accepted, at least for the time being.

An interesting difference between explicit and implicit acceptation is their consequence. In games where players have the opportunity to accept explicitly, statements are fixed in the game after acceptation. In case acceptation is implicit, the agreement is temporal: a player can always retract his agreement.

40 If after the move 'why P?' the statement 'Q' follows, a consequence is that both players are committed to the material implication 'if Q then P'.

Retractions
The logicians' conditional of MacKenzie's game, and the defaults in Brewka's game, cannot be retracted. Both games allow withdrawal of statements, by way of withdrawal and remove respectively. In DiaLaw withdrawal is allowed only if the player himself is committed and the other is not. In MacKenzie's game a player can withdraw even if he himself is not committed. In that case the move is meant to state explicitly the lack of commitment. In Brewka's game and DiaLaw, the meaning of withdraw is exclusively to end actual commitment.

Rescher's game and the Pleadings Game have no move by which statements can be withdrawn.

Questions
In DiaLaw question is the move by which a player asks his opponent to support some statement. Brewka's empty move is similar to question, and Gordon's deny can be considered as a question. The opponent is asked to support his statement, or, in terms of the Pleadings game, asked to adduce an argument that attacks the denial.

MacKenzie has three question-like moves. Similar to the ones discussed above is challenge. Special moves are question\(^{41}\) and resolution demand. With the first a player wants to know the opinion of the opponent about a particular statement. The resolution demand is used to confront the opponent with inconsistencies in his commitments.

Rescher's game has no questions. However, if the proponent claims a statement, and the opponent claims that the negation of the statement also seems to hold, this latter move by the opponent is comparable to a question. Namely, the consequence of this move is that the proponent has to adduce supporting evidence for the statement.

A table with all moves
This table provides an overview of all the different moves discussed. Concerning a particular statement there are three possible opinions:

1. convinced of a statement: A
2. convinced of the opposite: ¬A
3. not convinced of a statement or the opposite: A?, ¬A?

In the table is indicated whether one of the three opinions can be expressed by a certain move.

<table>
<thead>
<tr>
<th>claim statement</th>
<th>MacKenzie</th>
<th>Rescher</th>
<th>Brewka</th>
<th>Gordon</th>
<th>DiaLaw</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Statement (1,2)</td>
<td>Cat. ass. (1)</td>
<td>Add (1,2)</td>
<td>-</td>
<td>Claim (1,2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cat. ass. (3)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>claim rule</td>
<td>Statement</td>
<td>Prov. ass.</td>
<td>Add</td>
<td>Declare</td>
<td>Claim</td>
</tr>
<tr>
<td>claim argument</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Defend</td>
<td>Claim</td>
</tr>
<tr>
<td>accept</td>
<td>-</td>
<td>-</td>
<td>Concede (1)</td>
<td>Concede (1)</td>
<td>Accept (1)</td>
</tr>
<tr>
<td>withdraw</td>
<td>Withdrawal (3)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Withdraw (3)</td>
</tr>
<tr>
<td>ask support</td>
<td>Challenge (3)</td>
<td>-</td>
<td>Empty (3)</td>
<td>Deny (3)</td>
<td>Question (3)</td>
</tr>
<tr>
<td>ask opinion/</td>
<td>Question (3)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>resolution</td>
<td>Resolution demand</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

\(^{41}\) The name is confusing. Question has a different meaning than what normally is understood as question.
A claimed statement expresses the opinion that one is either convinced of the statement (1) or about the opposite (2). Only in Rescher's game the negation of a statement cannot be expressed, because of the asymmetric character of the game. The opponent's cautious assertion is not a positive claim, but, as the name suggests, a cautious one. Therefore, the reaction of the proponent to that claim is not a negation.

Neither the claim of a rule, nor the claim of an argument express an opinion. Rules are meant to express general relations between statements, while arguments express support for statements. It could be claimed that an argument expresses the opinion that one is convinced of the statement that the argument supports. However, this blurs the difference between a statement, and support for a statement.

If a player accepts a statement he expresses that he is convinced about a statement. In none of the games it is possible for a player to accept a statement if he is committed to the negation of this statement. Therefore the second opinion, convinced of the opposite, does not apply here.

Obviously, no opinion is expressed about being convinced if a player either withdraws (explicitly indicates to be no longer convinced), or questions (explicitly asking support, so not convinced yet).

### 6.2 Commitment

All moves discussed under the heading question, except challenge, do not influence the commitment of any player. Some moves do not influence the commitment of the opponent, such as claim in DiaLaw, remove and concede in Brewka's game and withdraw in MacKenzie's game.

<table>
<thead>
<tr>
<th>Commitment</th>
<th>Opponent</th>
<th>Loss of Commitment</th>
<th>Opponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacKenzie</td>
<td>Statement Challenge</td>
<td>Statement Challenge</td>
<td>Challenge Withdrawal</td>
</tr>
<tr>
<td>Rescher</td>
<td>Cat. ass.</td>
<td>Cat. ass.</td>
<td>Cat./Caut. &amp; Prov. ass. Caut. ass. Remove</td>
</tr>
<tr>
<td>Brewka</td>
<td>Add</td>
<td>Add</td>
<td>-</td>
</tr>
<tr>
<td>Gordon</td>
<td>All moves</td>
<td>Declare</td>
<td>-</td>
</tr>
<tr>
<td>DiaLaw</td>
<td>Claim Accept</td>
<td>-</td>
<td>Withdraw Accept Withdraw Accept</td>
</tr>
</tbody>
</table>

The moves in italic do not always have the described consequences. For instance, if a statement is challenged to which the opponent already was committed, he will, of course, not become committed; the challenging player does not lose his commitment if he was not committed in the first place. In DiaLaw accept only leads to loss of commitment if there are open statements that were claimed after the just accepted sentence was claimed.

---

42 The cautious assertion could have been placed in the question column, since the opponent asks evidence to defeat the cautious assertion. But because it is called an assertion, I prefer to consider it as the claim of a statement.
In the Pleadings game the moving player becomes committed after each move he performs: defend, concede, declare and deny. In Brewka's game the moving player becomes committed in case he adds or concedes a statement. In DiaLaw the moving player becomes committed after both claim and accept. In Rescher's game the proponent and opponent become committed to every categorical assertion. Because commitment in Rescher's game is defined as duty to defend, the players are not committed to the indisputable defaults.

Opponents only become committed to statements claimed by their adversary in games that do not have an accept move (MacKenzie, Rescher). In the other games explicit acceptation is needed, either called concede (Brewka, Gordon) or accept (DiaLaw).

After a challenge in MacKenzie's game the challenged player becomes committed, if he was not already. Another consequence is that the player who challenged becomes committed to his challenge.

In the Pleadings Game it is impossible to lose commitment after a move. In MacKenzie's game, DiaLaw and Brewka's game commitment ends after withdrawal. Furthermore, the player loses his commitment in MacKenzie's game if he challenges a statement he is committed to. In Rescher's game, the proponent who puts forward a provisioned assertion, loses his commitment to the statement that matches the conclusion of that default. The opponent loses his commitment to a statement if he attacks it.

In Rescher’s and Brewka’s game commitment is only changed as a consequence of the moves made by a player. Basically, the same holds for MacKenzie’s game. The only exception is the material implication to which both players become committed if a challenge is reacted to with a statement. Moreover, although players are not committed to consequences, they can be confronted with inconsistencies based upon such consequences. In the Pleadings Game players are committed to known consequences of their statements. DiaLaw uses forced moves to take care of what could be called strengthening of commitment. For instance, the player accepting the outweigh relation, is forced to accept the statement the relation pleads for.

7. OTHER RELATED RESEARCH

There is a wide range of other researchers who work(ed) on dialogical models, of which only a few will be discussed: Van Eemeren and Grootendorst, Vreeswijk, Nitta & colleagues, and Farley and Freeman.43

Van Eemeren and Grootendorst are leading figures in the field of (informal) argumentation, and their work inspired the development of DiaLaw. Vreeswijk [1993] was one of the first to develop a dialectical model in the formal argumentation field. From the first days that AI and Law became interested in dialogical models Nitta and others reported on the HELIC systems. Finally, Farley and Freeman’s [1996] well-known levels of burden of proof will be discussed.

---

43 Omissions are numerous, for instance from the AI and Law field are not discussed here, but certainly of interest: the work of Loui [e.g., 1992]; St-Vincent, Poulin & Bratley [1995], Prakken & Sartor [1996], Kowalski & Toni [1996]. The Zeno system [Gordon & Karacapilidis, 1997].
7.1 Van Eemeren and Grootendorst

Van Eemeren and Grootendorst [1982] co-authored a dissertation entitled *Rules for reasonable discourse*. Their aim was to develop a framework that could be used both as an instrument for analyzing discussions, and as a code of behavior for reasonable discussions. The framework was meant to deal with normal, daily life discussions between people. For that purpose they formulated general (universal) conditions that participants of a discussion must obey. An important starting point for their work is their opinion that communication has a social character. This does not necessarily imply that other people are directly involved in the communication. Even an interior dialog emphasizes the social character of argumentation, since in conducting such a dialog one anticipates possible reactions of others. Their theory of argumentation nowadays is known as pragma-dialectics [cf. Van Eemeren et al. 1996, p. 274].

A few key issues of their work are highlighted, and for each I indicate how DiaLaw copes with the issue. A fundamental condition for argumentation is:

A set of statements U1, U2, ..., Un is used by the speaker (S) to convince the hearer (L) of the (in)correctness of a statement M.

This condition is true for DiaLaw. In DiaLaw the speaker introduces the statements U1, U2, ..., Un one by one. If, for instance, U1 suffices to convince the hearer, the speaker stops; if U1 is doubted by the hearer, U1 becomes - for the time being - the new statement the hearer must be convinced about.

Apart from the fundamental condition, both a preparatory and a sincerity condition for argumentation limit the behavior of the speaker in discussions.

- **Preparatory conditions - the speaker believes that the hearer:**
  1. does not (yet) accept a statement M;
  2. accepts the set of statements U1, U2, ..., Un;
  3. accepts the set of statements U1, U2, ..., Un as a justification for M.

If the preparatory condition is not satisfied, then argumentation is considered inefficient. According to this condition DiaLaw is inefficient. If a player in DiaLaw believes that his opponent will accept a statement (i), he may nevertheless claim it. I do not consider this inefficiency as a drawback. If his opponent does not accept the statement after all, nothing is lost in terms of efficiency. If the opponent accepts the statement, at least it is explicitly clear that there is agreement. To me this explicit agreement is more important than possible inefficiency. The other two conditions are also different in DiaLaw. A player in DiaLaw just hopes, in stead of believes, that the hearer accepts the statements (preparatory condition 2) or the statements as justification (preparatory condition 3).

- **Sincerity conditions - the speaker believes that the following is acceptable:**
  1. M;
  2. U1, U2, ..., Un;
  3. U1, U2, ..., Un as a justification for M.

If the sincerity condition is not satisfied, then argumentation is considered untruthful. However, in practice it is impossible to check whether a speaker believes what he claims,

---

44 The thesis was in Dutch: *Regels voor redelijke discussie.*
because belief is a private matter. Therefore the sincerity condition is not part of DiaLaw. DiaLaw only deals with commitment. And although in general a player will believe what he is committed to, this does not necessarily have to be the case.

The conditions discussed so far apply at the most general level of argumentation. On a less general level eight preliminary conditions for a reasonable discussion are defined, such as: a speaker says what he means and is committed to what he says (3), a speaker is prepared to defend all his justifying and enfeebling statements (8), etc. In DiaLaw players are also committed to what they say, but they do not always have to be prepared to defend every statement they claim. If a player is not prepared, he is allowed to withdraw the statement.

The work of Van Eemeren and Grootendorst had great impact on the field of argumentation. In later work the framework was used to define argumentation guidelines for law students, and also to analyze different aspects of legal reasoning, such as the procedures in civil and criminal law [Feteris, 1989], analogy and e contrario argumentation [Kloosterhuis, 1996], judicial decision making [Plug, 1996], etc.

7.2 Vreeswijk’s argument-based dialectics

After nonmonotonic logics were introduced in a special issue of Artificial Intelligence in 1980, many new nonmonotonic logics have been developed. Generally characterized, these logics aim to model human reasoning by using theories of defeasible knowledge. From more recent date are the argument-based approaches, known as formal argumentation45 [e.g., Simari & Loui, 1992; Gordon, 1993b; Vreeswijk, 1993; Pollock, 1994; Dung, 1995; Prakken & Sartor, 1995; Verheij, 1996]. In these approaches the notions of argument and counterargument are central. Put simply, in the argument-based theories a conclusion is justified only if it is supported by an undefeated argument.

The reason to discuss the work of Vreeswijk is that his dialectical model of argumentation was one of the first fully developed argument-based dialectical models. Moreover, although Vreeswijk was not the first to use levels in argumentation, it was his work that inspired the way levels are used in DiaLaw.

Vreeswijk developed his dialectical system as a way to search a large data base of arguments. By way of a dialectical protocol, that defines how arguments pro and con a certain conclusion must be evaluated, Vreeswijk tried to diminish the computational problems normally faced when deriving whether a conclusion holds given a particular theory. Only arguments relevant for proving the main conclusion are evaluated.

The particular sequence in which the arguments are considered, influences the final result of the reasoning process, because “the order and direction in which arguments are developed, i.e. the procedure, is important” [Vreeswijk, 1993, p. 125]. An additional advantage of the dialectical approach is “that it deals with arguments in the original manner—as we do ourselves” [Vreeswijk, 1993, p. 111].

The present discussion focuses on how the arguments are exchanged.

---

45 As opposed to informal argumentation, e.g., [Van Eemeren, Grootendorst & Kruiger, 1987].
7.2.1 The debate in main lines

The dialectical protocol defines an artificial debate. The debate is called artificial because the parties in the debate are not supposed to be humans (as opposed to for instance the Pleadings Game and DiaLaw), but rather two separate computer programs playing against each other.\textsuperscript{46} The purpose of the debate is to test whether some thesis is in force given a particular background. This background is fixed at the beginning of the debate, and during the debate no new information may be added. The debate starts if the proponent puts forward the main thesis. During the debate the proponent adds arguments that (in)directly support the main thesis, the opponent submits arguments that (in)directly defeat the main thesis.

Each dispute is played against the background of an argumentation system. The argumentation system consists of a language \( L \) and a set of rules \( R \). The rules can be both strict and defeasible. Finally, there is a shared basis \( E.\textsuperscript{47} \)

A strict rule is represented as \( a \rightarrow b \), a defeasible rule as \( a \Rightarrow b \). The condition of the rule may be empty, e.g., \( \rightarrow b \) (b is always true), or \( \Rightarrow b \) (b is true as long as there is no counter evidence). Alternative conditions are not allowed, conjunctive conditions are. The rule \( (a \lor c) \rightarrow b \) is represented as \( a \rightarrow b \) and \( c \rightarrow b \). A rule may have only one conclusion. For instance, the rule \( a \rightarrow (b \land c) \) is represented as \( a \rightarrow b \) and \( a \rightarrow c \). Contradiction is denoted by \( \bot \). The conclusion of a strict rule can be a contradiction, e.g., \( a, b \rightarrow \bot \).\textsuperscript{48} This rule means that \( a \) and \( b \) together cause a contradiction.

Except in the first move, an argument must defeat the argument on the previous level\textsuperscript{49} by:

1. Causing a contradiction

Suppose that at a certain stage in the debate the conclusion \( b \) is justified, and \( a \) is in the shared basis. In case the strict rule \( a, b \rightarrow \bot \) is part of the set of rules, then \( a \) is a defeating argument, because this proposition causes a contradiction.

2. Supporting a proposition opposite to the one on the previous level

The argument \( a \Rightarrow b \) can be defeated by the argument \( c \Rightarrow \neg b \).

Note that the defeating argument \( c \Rightarrow \neg b \) does not cause a contradiction, because the argument \( a \Rightarrow b \) is defeated.

The arguments can be ordered in any way. A basic ordering is that arguments based on strict rules defeat arguments based on defeasible rules. A special ordering is for instance \( (a \Rightarrow b) \leq (c \Rightarrow \neg b) \), meaning that the first argument is defeated by the second. If there is no ordering on arguments, defeasible arguments with opposite conclusions defeat each other.

---

\textsuperscript{46} In later work Vreeswijk [1994] implemented the ideas described in his dissertation.

\textsuperscript{47} Vreeswijk denotes the shared basis as \( P \). To prevent confusion with \( P \) which I use to denote the proponent (Vreeswijk uses \( \mathcal{P} \)), I refer to the shared basis as \( E \), commonly used to denote evidence.

\textsuperscript{48} The comma denotes a conjunction.

\textsuperscript{49} The meaning of levels is explained later.
7.2.2 The moves

The game allows the following moves.

1. stating theses;
2. posing questions;
3. giving answers.

If a thesis \( \tau \) is stated, it is always accompanied by an exclamation mark, \( \tau! \). A thesis is always an argument.

The player who poses a question wants his opponent to confirm that an argument contains a particular subargument. The move, \( \sigma \subseteq \tau \), means: "do you admit that \( \sigma \) is a subargument of \( \tau \)?" The reason to perform this move is to commit \( \sigma \) explicitly to some subargument of his thesis. After the explicit commitment, normally the subargument is attacked.

The answer move is obligatory. After a question a player must react with an answer in which he confirms that some argument is indeed a subargument. An answer has no additional signs, and is a repetition of the subargument.

Consider an argumentation system with the following rules:51

\[
\{ \text{killer} \Rightarrow \text{murderer, self\_defense} \rightarrow \text{not\_punishable} \} \cup \\
\{ \text{not\_punishable, murderer} \rightarrow \bot \}.
\]

Suppose there is a basic ordering on arguments, so an argument based on a defeasible rule is defeated by an argument based on a strict rule. The set of rules contains the defeasible rule that killers are murderers; the strict rule that someone who acts in self defense is not punishable; the strict rule a contradiction is caused if someone is both a murderer and not punishable. If the shared basis \( E \) is \( \{ \text{killer, self\_defense} \} \), the following debate is possible.

P: killer \( \Rightarrow \) murderer!
O: self\_defense \( \rightarrow \) not\_punishable!

The defeasible rule is defeated by the strict rule (according to the basic ordering on arguments), and the proponent loses the debate. Another possible debate is:

P: self\_defense \( \rightarrow \) not\_punishable!

Because the proponent starts with the argument based on a strict rule, the opponent cannot put forward a counterargument and loses the debate.

7.2.3 The layered debate

A debate always starts when the proponent states the main thesis. The initial level of a debate is 1. So the argument by which the proponent starts the debate is stated on level 1. If the opponent states a thesis, the level becomes 2. In general, each time a thesis is stated, the level becomes lower. The debate returns to a higher level only if a thesis is conceded. A

---

50 The meaning is similar to Rescher's '!'.

51 The language of the argumentation system in this example is: \( \{ \text{killer, murderer, self\_defense, not\_punishable} \} \cup \{ \bot \} \)

52 I call this the highest level. The level 2, 3, etc. are lower levels, see also Chapter 3.
concession is not a separate move. If a player does not react to the latest argument, but to
one on a higher level, it is assumed that he concedes the lower level argument.53
The following picture gives an overview of all possible moves.

The picture completely defines the possible sequences of moves.
We start with the emphasized argument. A player can react to an argument in basically
four different ways. First, he can attack the argument. In that case he puts forward an
argument at level n+1. Secondly, he can ask whether the argument contains some sub-
argument. In that case the level remains the same. Finally, there are two different ways the
argument can be conceded. Either by questioning a two levels higher (or a multiple of two)
argument, or by attacking such a higher level argument.54
The arrows show that there is only one possible reaction to a question, namely an
answer.

After an answer there are three possible reactions. The only reasonable one is that the
subargument is attacked, i.e. that an argument is put forward at level n+1. The other two
reactions are highly unlikely and therefore indicated by dotted lines. Why pose a question
(to which the answer is an obligatory reaction) if the argument is conceded after all?
In general, an arrow that points down means that a counterargument is put forward. A
horizontal arrow means that the level remains the same. An arrow that points up means that
an argument is conceded. Note that in case the level n argument is the first of the game, the
upper part of the picture does not exist.

7.2.4 Other properties of the debate
In case there are only defeasible arguments concerning some proposition, and no priority
relation is defined, then irrespective of the number of arguments that plead for or against
the proposition the player who starts always wins.55 However, it may take some time.

53 In the example 6.3 (Vreeswijk 1993, p. 117) a concession is described as follows. "In line 5 P
immediately rebuts the argument of O in line 4. Because O comes down two levels in line 6, this is
with success."

54 The symbol k in the picture is odd. If one argument is conceded k=1. The level is n-1 in case of a
new counterargument, and the level is n-2 in case of a question. If more than one argument is
conceded k = 3, 5, 7, ... If n ≤ 2, then conceding, so the upwards arrow, means that the debate ends.

55 Because the starting player can always defeat an argument B that defeated his argument A with the
same argument A.
Suppose there is one argument for some proposition and ten against. If the game starts with the one argument for, the debate can last 21 moves. The argument pro is defeated by the first argument against in the second move, by the second argument against in the fourth move, and finally by the tenth argument against in the twentieth move. After adding the one argument pro in the twenty-first move, no counterargument is possible anymore. All counterarguments were already used to defeat the argument pro, and a counterargument may be used only once to defeat the same argument. If the game had started with one of the ten arguments against it could already be finished after three moves:

\[ P: \quad a \Rightarrow \neg b \]
\[ O: \quad c \Rightarrow b \]
\[ P: \quad a \Rightarrow \neg b \]

In case the proponent had put forward another argument in the third move, e.g., \( d \Rightarrow \neg b \), the debate would proceed, ultimately for a total of 21 moves.

A different approach is taken in DiaLaw, where reasons can outweigh one reason, but also one reason can outweigh ten reasons. It depends on the arguments supporting the particular weighing relation which set of reasons wins, and not on which player started.

**Termination**
If a party cannot adduce a defeating argument, a debate is won on substantial grounds. If a party is able to respond, but nevertheless does not so (e.g., because time is up), a debate is won on procedural grounds. Because the number of a move by the proponent P is always odd, and the number of a move by the opponent O is always even, a debate with an odd number of moves is won by the proponent, a debate with an even number of moves is won by the opponent.

**Relevancy**
In order to focus the dispute, the notion relevancy is defined, in a way comparable to Gordon's relevant statements. Arguments irrelevant for proving the main thesis are not allowed. Relevant to the main thesis are all arguments that (in)directly influence whether the main thesis is justified (or not).

The arguments are divided in a set of open arguments and a set of closed arguments. An argument is open as long as there is a counterargument possible. The set of closed arguments is divided into a set of established arguments and a set of refuted arguments. If an argument is ultimately defeated it becomes part of the refuted set, if an argument has survived all possible counterattacks it becomes part of the established set.

### 7.2.5 Concluding remarks
Because answers are obligatory, the question/answer-moves are superfluous. Instead, stating a thesis could be extended by indicating what argument is attacked by the move. If

---

56 Vreeswijk [1993, p. 116]: "... gave an answer to which the other party was not able to respond." This description is confusing, since, although a debate can end after an answer is given, it will normally end after an argument has been put forward. For instance, all the example debates in Vreeswijk's dissertation end after a thesis is stated (so after an argument is given).
the complete argument is attacked, nothing has to be indicated. If a subargument is attacked, the player must indicate what subargument he attacks.

The argumentation system is fixed, so all debates are finite. If the aim is testing conclusions in an efficient way this is important, for infinite debates are as inefficient as can be: everlasting. Apart from some inefficiencies described above, the debate is efficient because players only make relevant moves.

Because levels are used to structure the debates and the ordering on arguments is not necessarily based on specificity, but can be based on any ordering, Vreeswijk’s defeasible dialectics is an improvement of, for instance the approach of Rescher.

A strong point of abstract approaches like Vreeswijk’s is that the freedom in language and derivation makes it possible to model almost any debate. However, a drawback is related to this advantage: the modeler of the argumentation is hardly guided in his modeling activity. Therefore these systems are primarily suited to study aspects of argumentation in general, and less so for modeling commonsense reasoning in a particular domain.

7.3 The HELIC systems

The HELIC-II system [Nitta, Wong & Ohtake, 1993], later called new HELIC-II [Nitta et al., 1995; Nitta & Shibasaki, 1997], is a computational model for trial reasoning. The legal domain is that of Japanese criminal law.

While in DiaLaw and the Pleadings Game the debate is between humans, and the participants in Vreeswijk’s dialectics are two computers, the debates of the HELIC systems take place between a human player and the computer. A debate starts when either the computer or the player makes an argument. The player who begins is the prosecutor, the other player is the defendant. In the course of the debate the players exchange arguments as follows.

Each argument is based on a rule, which can be any rule: a legal rule, a commonsense rule, a rule derived from case law, etc. The computer automatically generates the arguments from a prefixed set of rules. Because the rules in new HELIC-II are given, it is not possible to argue about their validity.

The system not only generates arguments, but also provides the player with the best possible argument. The player only has to indicate his goal (conclusion of the argument), and his viewpoint (either prosecutor or defendant). A goal corresponds to one of the issues of the other player’s argument. It is the system that derives a list of issues from the other player’s argument. It is the player’s task to select the best one to attack. So the role of the player is restricted. In DiaLaw a player can adduce arguments, claim statements; in the Pleadings Game a player can declare rules and adduce arguments. In new HELIC-II he can only select issues. If a player selects an issue and the system cannot create a counterargument, the player loses the discussion about the issue.

What is the best argument in a particular case is determined in two distinct steps. First, the argumentation module creates all possible arguments given the goal of the player. For instance, if the player’s goal is to establish that O.J. is a killer, the system generates all arguments based on rules that have as conclusion that someone, in this case O.J., is a killer. Also the system generates all counterarguments, countercounterarguments, and so on, if there are any. Subsequently, depending on the viewpoint of the player the best argument is selected. For instance, if the viewpoint is that of the prosecutor, the argument is selected that supports that O.J. is a killer in the best possible way, also regarding possible
counterarguments. The ordering of arguments is based on the priority relation between the rules.

An argument can be countered by only one argument at the time. It is not possible to put forward more than one argument at a time and as a consequence comparison between arguments only takes place one by one. However, in case an argument is withdrawn, another, new argument can replace the previous one. So, there can be more than one argument for the same issue in the course of the debate, but not at the same time. Arguments can be countered in depth as long as there are counterarguments available. A counterargument can be countered by a countercounterargument, the latter by a countercountercounterargument, etc.

Each time there are arguments for and against an issue a player can claim that his argument is stronger. If his opponent accepts this claim or if he accepts the counterargument, he loses the issue, or, if there are still other supporting arguments, he can put forward a new argument. If there is an argument pro and con, probably both players will claim their argument is stronger. In that case the relation between the arguments remains an irresolvable issue. A move similar to arguments for an outweigh relation in DiaLaw, is not possible.

The debate only ends if the initial argument is accepted, or more correctly, if the defendant indicates that he no longer knows a counterargument against it. A consequence of this definition of the end of the debate is that it is impossible for the prosecutor to lose. What happens if the prosecutor accepts that the argument of the defendant defeats his own is not clear. One would expect the prosecutor to lose, but according to the game’s definition the debate can only end if the initial argument is accepted.

There is a drawback related to the fact that HELIC is a closed system, while the law is an open system. A player may lose a debate even in case he has strong arguments, simply because the system does not produce those. Temporarily this will remain a problem for legal argument games between humans and computers, because if a player is allowed to enter moves freely, the computer probably will not understand every move.

On the other hand, the system has strong points too. First, the system uses a rich language. It would have led me too far astray to discuss this language, but for those interested in legal knowledge representation it is worthwhile to study it. Secondly, if a player has made a bad argument, he still has the chance to come up with a better one.

7.4 Farley and Freeman’s levels of burden of proof

From a legal point view a distinction between different levels of burden of proof is interesting. Farley & Freeman [1995; Freeman & Farley, 1996] elaborate on this distinction.

Arguments are modeled in what they call a Toulmin Argument Unit. The warrant of an argument has two type fields. The first type classifies the relationship between the condition and the conclusion of a warrant either as explanatory or as indicating a correlation. The second type represents with which strength a conclusion may be derived using the warrant:

1. sufficient - strength is conclusive, e.g., definitions;

---

57 See chapter 2.
2. default - usually, almost always;
3. evidential - less certain, still likely.

The strength of an argument depends on the strength of the warrant type (1-3), and which reasoning step is used. The reasoning steps they distinguish are Modus Ponens (MP), Modus Tollens (MT), Abductive (AB) and Contraposition (CO).  

The strength of an argument is either unknown, or, in order of diminishing strength, valid, strong, credible or weak. The strength of an argument can be determined as follows. If the reasoning step is MP, then the argument is valid if the warrant type is sufficient; the argument is strong if the warrant type is default, and the argument is credible if the warrant type is evidential. Irrespective of the warrant type, an argument using either CO or AB as a reasoning step is always weak. Finally, an argument built using MT is valid if the warrant is sufficient, and otherwise it is weak. The following table presents an overview of the strength of arguments.

<table>
<thead>
<tr>
<th>Sufficient warrant</th>
<th>Default warrant</th>
<th>Evidential warrant</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP Valid</td>
<td>Strong</td>
<td>Credible</td>
</tr>
<tr>
<td>MT Valid</td>
<td>Week</td>
<td>Weak</td>
</tr>
<tr>
<td>AB/CO Weak</td>
<td>Weak</td>
<td>Weak</td>
</tr>
</tbody>
</table>

The strength of arguments is used to distinguish between different levels of support, or, in other words, different levels of the burden of proof. For each level holds that the argument used must be defendable, which means that it cannot be defeated (e.g., on the basis of specificity). The following classification of levels of support can easily be applied to one’s own favorite dialectical argument system.

1. Scintilla of evidence
   At least one weak, defendable argument (AB, CO, and MT with a default or evidential warrant), regardless of support for any rival claim.

2. Preponderance of evidence
   At least one weak, defendable argument outweighs the other side’s argument (see 1).

3. Dialectically valid
   At least one credible, defendable argument defeats all of the other side’s arguments (MP with a evidential warrant).

---

58 Given the implication p → q, the different reasoning steps are:

<table>
<thead>
<tr>
<th></th>
<th>Given</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modus Ponens</td>
<td>p</td>
<td>q</td>
</tr>
<tr>
<td>Modus Tollens</td>
<td>~q</td>
<td>~p</td>
</tr>
<tr>
<td>Abductive</td>
<td>q</td>
<td>~p</td>
</tr>
<tr>
<td>Contraposition</td>
<td>~p</td>
<td>~q</td>
</tr>
</tbody>
</table>

59 An important aspect of the burden of proof is which party buries it. For Farley and Freeman it is always the side who made claim that has the burden to prove it.
4. **Beyond a reasonable doubt**
   At least one strong, defendable argument defeats all of the other side’s arguments (MP with a default warrant).

5. **Beyond a doubt**
   At least one valid, defendable argument defeats all of the other side’s arguments (MP, MT with a sufficient warrant).

Maybe other types of reasoning, such as inductive reasoning, should be accounted for, but this can be done as far as it is necessary for a particular argumentation system, or a specific domain.

Although most legal rules will be of the default type, it is clear that it will not be an easy task to assign the appropriate strength to warrants. In what cases a warrant is evidential or sufficient is not easily determined. Correct assignment, however, is important, because wrong assigned values may cause false conclusions.

Regardless of this difficulty, the division in different levels of support for legal applications seems fruitful. For instance, differences in levels of support for claiming someone’s guilt in Criminal Law (“legally and convincingly proved”) and for validating claims in Civil Law (“plausible”) can be accounted for.

However, there is yet another reason to be cautious. If warrants are labeled ‘default’, as most legal rules will be, it is impossible to prove something beyond a doubt. So in that case, a burden to prove something beyond a doubt cannot be demanded. These difficulties must be recognized.

It is not claimed that the levels of support are only these five, neither that the warrant strengths are the three mentioned, nor even that the reasoning steps are restricted to the four described. All these distinctions can be adapted to one’s own wishes. If one is cautious, the basic principle to determine levels of support is a useful one.

8. **Conclusion**

Three models have been discussed extensively, MacKenzie's DC, Rescher’s Dialectics (and Brewka’s formalization), and Gordon’s Pleadings Game. An overview of both moves and commitment was given. Since each discussion ended with concluding remarks, the differences between the systems are sketched here only roughly.

The statement-based dialogs in MacKenzie’s DC are natural even if the system is implemented, but the use of indefeasible rules is a drawback if the domain of law is modeled. The dialogs in Rescher’s system are also natural. Although defeasible rules are used, a drawback is that defeat is only based on specificity, and the rules are indisputable. Gordon’s Pleadings Game uses a rich language especially suited for the legal domain and he uses defeasible rules. However, the resulting dialogs are not natural, and drawbacks of his system are that arguments may only be real arguments, and that only one argument supporting the same statement may be adduced. The dialogs in Dialaw are not as natural as in MacKenzie’s DC, but the just mentioned disadvantages are not present in Dialaw. A drawback of Dialaw is, for example, that dialogs cannot be stopped without losing them; a problem that is elegantly solved in the Pleadings Game.

---

60 For instance, in Freeman [1994] *beyond a doubt* was not a category.
The negation claim in DiaLaw is a powerful way to get the initiative in a dialog. Since Rescher's dialectics is an asymmetric game, the initiative is in hands of the proponent all the time. In DC a discussion about the negation of a claim is impossible if the opponent makes new claims to which the proponent automatically is conceded. In the Pleadings Game a player who has evidence that a statement is not true, may not put forward his evidence immediately, but first has to ask his opponent to prove that the statement is true.

All discussed models have strong and weak points, all models have their merits. What the models share is the basic idea of how to establish justified statements, namely by way of a dialectical exchange of statements and arguments. It is the dialectical aspect that is important if argumentation, especially if legal argumentation is modeled.

Future models may improve existing models, but it is necessary that these models stick to the dialectical principle. Only if the dialectical process is modeled, models of argumentation in general, and of legal justification in particular, can be true models, namely models that reflect the spirit, the essence of argumentation.
Chapter 6

What is an argument?
The logical and psychological perspective

We've drifted past the point of reasons why.

The starting point of my research was the observation that by convincing arguments an ostensibly lost case can be won. We are now in the position, after the introduction of the model DiaLaw and the discussion of other dialogical models of argumentation, to elaborate on the notion of an argument.

The chapter is structured as follows. First, argumentation is discussed from two different angles: the logical perspective and the psychological perspective. Two types of argumentation that are related to the two perspectives are discussed next: rational and a-rational. In the context of the present chapter the latter type of argumentation will be called dia-rational. Subsequently it is argued that a dialogical model should allow for both modeling rational as well as (d)ia-rational argumentation. To my knowledge, DiaLaw is the only model in which both the logical and psychological perspective are represented.

1. Argumentation: Two Perspectives

It is commonly accepted that one goal, maybe even the main goal of argumentation is to justify conclusions. In this section the way this goal can be realized is analyzed from two different perspectives. First, the logical perspective, and second the psychological perspective.

The logical perspective has a long tradition and still dominates the study of argumentation. The central question is what requirements argumentation must meet in order to guarantee that for someone who accepts the premises it is warranted to accept the conclusion.

In the psychological perspective argumentation is not analyzed in terms of logical compelling arguments. The central question is rather when argumentation is effective.

1.1 The logical perspective

In the logical perspective it must be guaranteed that if the premises are accepted, it is warranted to accept the conclusion. The obvious example that meets the requirements is the Modus Ponens argument.

\[
A \rightarrow B \\
A \\
--- \\
\text{therefore } B
\]

1 Lyrics from the 1984 album *Fugazi* by Marillion.
If someone accepts that $A$ is the case, and he accepts that $A \rightarrow B$ is the case, it is warranted for him to accept that $B$ is being the case. Note that the Modus Ponens argument has three elements. First, a set of premises: $\{A \rightarrow B, \ A\}$. Second, a conclusion: $B$. Finally, a derivation rule (implicitly present in the Modus Ponens argument), that guarantees that if the set of premises is accepted, it is warranted to accept the conclusion.\(^2\) If $\{A \rightarrow B, \ A\}$, then $B$.

The notion of an argument from the logical perspective is also present in modern theories of defeasible argumentation. Examples are the arguments in the already discussed models of Gordon [1995], Vreeswijk [1993], and Nitta et al. [1995]. Put simply, in all these models an argument is based on the application of a rule. The difference with the Modus Ponens argument is that a conclusion is only derived provisionally. New information can change the status of the conclusion. If a conclusion is derived by the Modus Ponens argument, the conclusion stands once and for all.

In the logical perspective procedures play no role. In case of a logically valid argument there is a criterion by which can be determined whether the conclusion follows from the premises. This criterion does not refer to a procedure. This does not preclude the possibility that arguments are generated in a procedure, but this procedure is irrelevant to the justifying force of the arguments.

Although in the models of Gordon and the others the arguments are defined to be used in a procedure, the definition of the arguments is still independent of the procedure. Just like the Modus Ponens argument is defined independent of a procedure, but can nevertheless be used in a procedure [see Lorenz, 1961; Alexy, 1989].

The following example based on DiaLaw illustrates that an argument that is defined independently from a procedure, i.e. is an argument in the logical perspective, can nevertheless be used in a procedure. If someone accepts that $\text{outweighs}(\text{reason}, \ \{\})$, conclusion, then it is warranted for him to accept conclusion. The conclusion is justified irrespective of any procedure. In general, if there is a non-empty set of reasons for a conclusion and an empty set of reasons against a conclusion, it is warranted to accept the conclusion. This argument could also be used in a procedure, as it is in DiaLaw. Nevertheless, the validity of this argument does not depend on the use of the argument in DiaLaw.

1.2 The psychological perspective

The psychological perspective is one that was, to my knowledge, introduced by Stevenson [1944, p. 113]:

"The reasons which support or attack an ethical judgement (...) are related to the judgement psychologically rather than logically. They do not strictly imply the judgement in the way that axioms imply theorems (...)"

\(^2\) In Propositional Logic there are two equivalent ways to interpret the Modus Ponens argument [cf. Van Dalen, 1980] that are combined here. From a semantic perspective the conclusion of an argument is true if the premises are true (see also Chapter 2); from a proof-theoretical perspective the conclusion is derivable from the premises.
What is an argument? The logical and psychological perspective

In the logical perspective conclusions are justified by logically valid arguments, like theorems following from axioms. In the psychological perspective conclusions are justified by effective, convincing arguments. The structure of the argument is not important, but rather is its effect. This effect can only be measured if the arguments are really used, if some audience is convinced by the arguments. An argument is effective if it causes the audience to accept the statement that had to be justified.

Argumentation in the psychological perspective is in some respects the opposite of argumentation in the logical perspective. In the psychological perspective the structure of arguments is irrelevant and the argumentation has to be conducted in a procedure; in the logical perspective the structure of the arguments is important, and a procedure is only optional, not necessary.

2. ARGUMENTATION: TWO TYPES

Rational is what is considered rational by rational beings. I adhere to this minimalistic, almost circular definition of rationality. There is, however, the more broadly accepted view on rationality. In that view rational argumentation is connected to the logical perspective. Connected to the psychological perspective is what I call (di)ra-rationality.

Note that “the concept of rationality is (...) both ambiguous and applicable to vastly different areas.” [cf. Peczenik, 1989, p. 56]. The discussion of rationality (see also Peczenik [1989, p. 56f.]) in the present chapter focuses on the rationality of legal conclusions.

It may seem that the logical perspective represents what Aarnio [1987, p. 189] calls logical rationality (L-rationality), and the psychological perspective represents what Aarnio [1987, p. 190] calls discursive rationality (D-rationality). As far as L-rationality is concerned this is almost true. The only difference is the type of inference used. Aarnio demands a deductive inference, while in chapter 2 it was argued that legal justification should not be modeled using a deductive logic. Correspondingly, the inference of the logical perspective discussed here can be based on any, not only deductive, logic.

The D-rationality differs from (di)ra-rationality. While D-rationality is only about justifying premises in a rational discussion, the (di)ra-rationality discussed here is not only about justifying premises in a discussion, but also about justifying conclusions in a discussion, even if the conclusions are justified without using logically compelling arguments.

In this section I elaborate on what could be seen as rational, and what could be seen as (di)ra-rational argumentation. An example from case law is used to illustrate the need for models of argumentation that allow (di)ra-rational argumentation.

2.1 Rational and dia-rational argumentation

Argumentation is rational if the premises are sufficient to justify the conclusion. Whether the premises are sufficient has to be determined by generally accepted derivation rules: given the premises, the conclusion must be derivable. It is evident that a logic is an appropriate candidate for this task. As an example the Modus Ponens argument can be

---

3 Peczenik [1989, p. 57] adds a third category, S-rationality. This type of rationality takes into consideration the coherence of the premises. A conclusion is S-rational if it follows logically from a coherent set of premises.
mentioned. The underlying logic, Propositional Logic, defines that the premises of the Modus Ponens argument are sufficient to accept its conclusion. Whether the conclusion is actually accepted is irrelevant. So, in rational argumentation it is irrelevant whether the conclusion really is accepted, what only matters is whether the premises are sufficient to accept the conclusion.

For rational argumentation rules are essential. The arguments, or reasons that justify conclusions, are based on rules, or a rule can be constructed from the reason or argument that has been used to justify. The rule indicates from which premises a conclusion can justifiably be drawn.

Currently, most researchers on formal argumentation model arguments that are based on rules [e.g., Lin & Shoham, 1989; Vreeswijk, 1993; Gordon, 1995; Dung, 1995]. In all these approaches an underlying logic is used to define valid inference. Given premises, it can be determined whether a conclusion is acceptable.

Argumentation is (di) rationale if adduced premises cause acceptance of the conclusion without these premises being sufficient to accept the conclusion. In case of (di) rationale argumentation it is not exactly clear why the conclusion was accepted. The actual acceptance is essential, because the premises are not sufficient to accept the conclusion.

Rules can play a role in (di) rationale argumentation, but do not have to be used. An example of the use of a rule in (di) rationale argumentation is the following. The conclusion that Oedipus should be punished could be accepted only after the general rule that killers are punishable was adduced to support this conclusion. In that case the argumentation is not rational, because the premise is not sufficient to accept the conclusion. The conclusion also could have been accepted without a rule being adduced, e.g. after only the statement was adduced that Oedipus is a killer. The argumentation is still (di) rationale, because the premise is not sufficient to accept the conclusion.

2.2 (Di) rationale argumentation in case law

Not only the argumentation of academics or rhetorically well-equipped attorneys, but also the argumentation of courts is regularly a rationale. A general example is that a decision is often justified in the following way. A couple of statements is adduced and then the magic phrase "in view of the above" (in Dutch: gezien het bovenstaande) connects the conclusion with the previous statements.

Statement1
Statement2

Statement_n

"in view of the above"

Conclusion

Decisions of this type could best be modeled as (di) rationale argumentation. The court claims one or more statements, and subsequently assumes that it is self-evident that the conclusion follows from the statement(s). Interestingly enough, the argumentation works, the argumentation is effective; mostly the argumentation of the court is convincing, although it is a rationale.
By adding a premise the argumentation could be transformed into what is broadly accepted as rational argumentation. Often it is argued that such a premise is already implicitly present. For instance, if someone argues that the conclusion B follows given A, the premise $A \rightarrow B$ can be added. However, this reconstruction of the argument not necessarily models the actual argument. It is not always intended that the conclusion follows logically from the claimed statements. Instead, statements are claimed to convincingly show that the conclusion is justified (or true, correct). The statements are rhetorical means, they are meant to convince. The connection between the statements and the conclusion that is obtained by adding a premise, is useful for those that only believe in logically valid arguments. If the concept of argument is enlarged to include the rhetorical side also, it is better to leave out the additional premise. Adding the ‘missing’ premise does not really add something to the argument, except an often unnatural connection between a set of statements and a conclusion.

The following case is of a different kind (the phrase ‘in view of the above’ is not used), but is yet another example of the a-rational character of argumentation by the Dutch Supreme Court.

In the Dutch Road Traffic Act it is said that in case of an accident between a motorist and other non-motorizing road users, e.g., pedestrians or bikers, the motorist has to pay for the damages, except if he acted in force majeure. In the early nineties the Dutch Supreme Court decided that in case a child not older than fifteen is involved in the accident, the motorist always has to pay the damages, except when the child acted recklessly or intentionally.

In a later case the Dutch Supreme Court had to decide whether the rule about young children should also be applied to elderly people. The Dutch Supreme Court decided that the rule was not applicable, based on the following statements:

1. the arguments used to define the rule for young people are less striking for elderly people;
2. law knows no fixed age criterion for elderly people (for young people it does);
3. elderly people are less recognizable.

These three statements formed the reasons for the decision of the Dutch Supreme Court. This collection of statements, however, is in no way sufficient to accept the conclusion. It could be argued that the missing premise is a rule that can be constructed in order to make the premises sufficient to accept the conclusion. However, instead of creating a rather artificial rule, it is better to model the argumentation dialogically, where the conclusion is

---

4 See on this topic also the so-called ‘jumps’ of Peczenik [1989, p. 1151].
5 The concerning rule was at that time laid down in section 31 of the Road Traffic Act (art. 31 Wegenverkeerswet). In the current Road Traffic Act the regarding section is 185 (art. 185 Wegenverkeerswet 1994).
8 I do not claim that it is never possible to model the ratio decidendi of a case as a rule (see on this topic Loui & Norman [1995]; Prakker & Sartor [1997]).
accepted after only the statements are adduced. So, the argumentation of the Dutch Supreme Court could best be modeled in an (di)a-rational way, where the statements caused the acceptance of the conclusion.

3. TOWARDS COMBINING THE APPROACHES

Rational and a-rational argumentation are presented as two extremes. Because in practical argumentation often premises remain implicit, few argumentation could be considered rational according to the above description.

A task for theories of argumentation is to define what good argumentation is. Suppose we are in search of arguments justifying that Oedipus is a murderer. What can be considered an argument for this statement?

a. We all hate him;
  b. He cold-bloodedly killed his father;
  c. He cold-bloodedly killed his father, and those that shoot their fathers cold-bloodedly are murderers.

If Oedipus is considered a murderer only on the basis of the statement that we all hate him, the argumentation is obviously bad. The premise is, or at least seems, far from being sufficient to accept the conclusion. However, if Oedipus is seen as a murderer on the basis of the statement that he cold-bloodedly killed his father, the argumentation is rather convincing. Nevertheless, also in this case the premise is not sufficient to accept the conclusion. In, for example, Gordon’s Pleadings Game, this second statement would not be allowed as an argument, for only real arguments according to the underlying logic are allowed. This argument is not of that type since it is not based on the application of a rule. Only if the third statement is claimed, the premises are sufficient to accept the conclusion.

In Chapter 2 it was argued that legal justification should be modeled dialogically. I believe that current dialogical approaches tend to stick too much to the use of rules in argumentation. This holds even for the game of MacKenzie that is, other than most AI & Law models not argument-based but statement-based. Namely, in the game of MacKenzie the consequence of a statement that is adduced to justify a challenged statement is that the players not only become committed to the adduced statement, but also to the rule-like material implication that combines the adduced statement with the conclusion. In the present example this means that the argument that Oedipus killed his father in cold-blood, would automatically be accompanied with the statement “If Oedipus killed his father in cold-blood, then Oedipus is a murderer.”

A dialogical model is a procedure in which the effect of argumentation can be modeled. Therefore, a dialogical model is suited to represent the rhetorical side of argumentation; a dialogical model is suited to model (di)a-rational argumentation. Moreover, if a dialogical model uses an underlying logic, it is also possible to model rational argumentation. In the next section it is shown how the two types of argumentation are combined in DiaLaw.

---

9 Note that in Gordon’s new work on the Zeno system [Gordon & Karacapiligis, 1997] the argument can be modeled. He has left the rule-based argumentation behind.
4. **DIALAW: RATIONAL AND DIA-RATIONAL ARGUMENTATION**

In case of rational argumentation the sequence in which the premises were adduced is irrelevant. History is of no importance. What only matters is whether the premises are sufficient to accept the conclusion.

In case of (dia)-rational argumentation the sequence in which the premises were adduced is important. Since the premises are not sufficient to accept the conclusion, by analyzing the context in which the premises were adduced it can be understood why the conclusion was accepted.

Dialaw is a model that combines rule-based justification and justification that is not based on rules. It combines the rule-based rationalism discussed in this chapter, and the rule skepticism as expressed by the American Realists. Therefore I would like to call the theory underlying Dialaw *Rationalistic Realism*.

To my knowledge, Dialaw is the first game that combines rational and (dia)-rational argumentation, and Dialaw is the only model in which reasons can have underlying rules, but do not have them necessarily.

I will give an example of both rational and (dia)-rational justification in Dialaw.

If justification is rational, the premises are sufficient to accept the conclusion. Players playing Dialaw must obey the rules of the game, and these rules include elements from Reason-based Logic, that define certain premises as sufficient to accept a conclusion. In case a player, say Ernie, has accepted such sufficient premises, and his opponent, say Bert, claims or has claimed that the conclusion that follows from these sufficient premises holds. In that case Bert is forced to accept this conclusion. The following informal dialog is an example of rational argumentation in Dialaw.

**Bert:** Oedipus is a murderer
**Ernie:** Why?

(...) 
**Bert:** The reasons pro 'Oedipus is a murderer' outweigh the reasons con
**Ernie:** I accept that the reasons pro 'Oedipus is a murderer' outweigh the reasons con

*Ernie is forced to perform the next move*
**Ernie:** I accept 'Oedipus is a murderer'

Because Ernie accepted that the reasons for the conclusion outweigh the reasons against the conclusion, he is forced to accept the conclusion. Another example of rational argumentation is the case in which Ernie accepts that a rule applies. If Bert claims or has claimed the reason based on that rule, Ernie is forced to accept that reason.\(^{10}\)

If justification is (dia)-rational, a conclusion is accepted without the acceptance being forced. In that case it is not exactly clear why the conclusion was accepted. A trivial example of (dia)-rational argumentation is a statement that is immediately accepted. However, this could still be considered rational argumentation, because the premises and conclusion coincide in this case. The premises are the conclusion, so the premises are

---

\(^{10}\) More examples of forced acceptance, and examples of forced withdrawal can be found in Chapter 3.
sufficient to accept the conclusion. A conclusion is really justified (di)rationally, if it was questioned and there has been no premises adduced that are sufficient to accept the conclusion. The following informal dialog is an example of (di)rationally argumentation in DiaLaw.

Bert: Oedipus is a murderer
Ernie: Why?
Bert: He cold-bloodedly killed his father
   Ernie voluntarily performs the next move
Ernie: I accept ‘Oedipus is a murderer’.

The statement that Oedipus is a murderer is questioned. It is accepted after Bert claimed the statement that Oedipus cold-bloodedly killed his father. Although this statement is not sufficient to accept the conclusion, Ernie accepted the conclusion anyhow. So Ernie accepted the statement after (di)rationally argumentation only.

Statements are justified in DiaLaw if they are accepted. If a player can be forced to accept a statement, he has claimed or accepted previously in the game premises that are sufficient to accept the statement. This is an example of rational argumentation. A player can also accept a statement without being forced. In that case the argumentation is (di)rationally.

5. Conclusion

In legal justification both rational and (di)rationally argumentation play a role.

In a dialogical model the adduced premises do not have to be sufficient to accept the conclusion, because the effect of the premises can be determined in the dialogical procedure. So the modeled argumentation can be (di)rationally. However, if the dialogical model uses an underlying logic, it is possible to force the opponent to accept a conclusion. The opponent can be forced to accept the conclusion if he is already committed to premises that are sufficient according to the underlying logic. Therefore, the modeled argumentation can also be rational.

DiaLaw has an underlying logic that is based on Reason-based Logic. Therefore, in DiaLaw justification can be rational. However, because a statement can be accepted without the premises being sufficient to accept the conclusion, in DiaLaw it is also possible to justify statements by (di)rationally argumentation.
Chapter 7

In conclusion

And when at last the work is done
Don’t sit down it’s time to start another one

In the first chapter questions were raised; in this final chapter answers are given. Subsequently the contribution of this thesis is addressed, and followed by a few closing remarks, amongst other about directions for future research.

1. The answers

In Chapter 1 four questions were raised. I repeat them here and provide answers. Each answer is accompanied by a reference to the chapter where the subject was addressed.

The first question was: How can be determined whether a legal statement is justified, and are there criteria a legal statement must satisfy in order to be justified?

Answer: There are no standards or independent criteria by which it can be determined whether a legal statement is justified. Since this lack of an independent criterion a procedural, dialogical model of justification is proposed. A legal statement is justified in the dialogical model if it is agreed upon by the participants. Whether the justified statement is also just, is a question that cannot be answered (see Chapter 2).

The second question was: What model of legal justification is satisfactory, and is it possible to define and implement this model?

Answer: In Chapter 2 it was argued that legal justification should be modeled dialogically. To this purpose the dialog game DiaLaw was defined and implemented. The purpose of the game is that the players convince each other, by performing speech acts. The model combines exchange of statements and exchange of arguments. Levels structure the dialog, the commitments of the players are stored, and the dialog history can be depicted by means of trees (see Chapter 3, 4).

The third question was: How do comparable models represent argumentation?

Answer: Three open models of dialogical argumentation have been discussed extensively: Rescher's Dialectics, MacKenzie's DC, and Gordon's Pleadings Game. Moreover, a survey of moves and commitment of these models and DiaLaw was given. Other contributions to the field of dialogical argumentation were discussed: the work of the Dutch linguists Van Eemeren & Grootendorst, Vreeswijk's dialectics, the HELIC systems of Nitta & others, and Farley & Freeman's levels of the burden of proof (see Chapter 5).

The fourth question was: What is acceptable as an argument in legal justification?

Answer: Argumentation is rational, if the premises guarantee the conclusion. Argumentation is a-rational, referred to as (di)er-rational, if the conclusion is accepted without the premises being sufficient to accept the conclusion. A dialogical model must allow both types of argumentation (see Chapter 6).

1 Lyrics from the 1973 album Dark side of the moon by Pink Floyd.
2. THE CONTRIBUTION OF THE THESIS

A theoretical foundation for a dialogical model of legal justification was given. After discussing the differences between the product and process of justification, three objections against models of justification were discussed.

First, the well-known objection against particularly a deductive model was addressed: the defeasible nature of legal justification. Second, it was argued that because of the open nature of law, a model of legal justification should be open. Finally, the Münchhausen Trilemma was presented as an objection against any model of justification. If justification is based on 'justified' premises, these premises are either part of a circle, or dogmas. If statements are justified in an open system, justification has no natural end, and therefore can go on infinitely.

To deal with the objections, a dialogical model of justification was proposed. Basically, a dialogical model of justification can be characterized as follows:

- there are two participants;
- rules regulate the exchange of arguments between these participants;
- statements are justified if the participants agree upon them.

If and only if a statement is accepted in a dialog game, the statement is justified. What is justified only holds for the participants in a particular dialog. Moreover, statements that are justified in one dialog can be rejected in a new dialog. So justification is not only audience-dependent, but also time-dependent. Justification of a statement for all, or for all rational beings, is impossible in my opinion.

In Chapter 2 it was claimed that:

- the open nature of law is an argument against non-procedural models of justification;
- although an independent criterion to decide whether a statement is justified does not exist, and whether a statement is justified can only be determined in a procedure, it is nevertheless impossible to define a procedure that guarantees that all statements that are justified in the procedure are just.

The dialogical model DiaLaw has been developed and implemented in Prolog as a prototype (see Appendix). Whether sentences have to be justified depends on the reaction of the other player; DiaLaw is a reaction-based game. A claim can only be guaranteed to be a reaction if there is a link with a previous move. This relation is guaranteed: a claimed sentence is either a reaction to a demand for justification (question) or a reaction to a claimed sentence (denial).

The language of the game is based on Reason-Based Logic [Hage & Verheij, 1994].

By way of the special language elements that are primarily based on this logic [Lodder & Herczog, 1995] the opponent can be forced to accept or withdraw. However, a player can only be forced if both players agree at least on some points.

The layered structure of the dialogs provide a good insight in the argumentation, most prominently by way of the dialog trees. To my knowledge, DiaLaw is the only model in which arguments, i.e. reasons, can have underlying rules, but do not have them necessarily.

---

2 For recent developments of Reason-based Logic see Hage [1996, 1997a] and Verheij [1996].
In conclusion

Two legal cases, the Tyrell case and the Chabot case, have been modeled in DiaLaw. The resulting dialogs illustrated the definitions and rules of DiaLaw, and showed extensively how legal justification can be modeled in DiaLaw.

Three open models of argumentation have been evaluated extensively. As an introduction to these models the same dialog was modeled in each of these models. In general terms, the models can be characterized as follows.

The statement-based dialogs in MacKenzie's DC are natural even if the system is implemented, but the use of indefeasible rules is a drawback if the domain of law is modeled.

The dialogs in Rescher's system are also quite natural. Although defeasible rules are used, a drawback is that defeat is only based on specificity, and that the rules are indiscutable.

Gordon's Pleadings Game uses a rich language especially suited for the legal domain and he uses defeasible rules. However, the resulting dialogs are not natural, and drawbacks of his system are that arguments may only be real arguments, and that only one argument may be added supporting the same statement. The dialogs in DiaLaw are not as natural as in MacKenzie's DC, but disadvantages of the other models are not present in DiaLaw. A drawback of DiaLaw is that dialogs cannot be stopped without losing them, a problem that is elegantly solved in the Pleadings Game.

Elements of the models by Van Eemeren & Grootendorst, Vreeswijk's dialectics, the HELIC systems of Nitta & others, and Farley & Freeman's levels of burden of proof were discussed and compared to DiaLaw.

A distinction was made between two perspectives in argumentation: the logical and the psychological perspective. In the logical perspective conclusions are justified by logically valid arguments, like theorems follow from axioms. In the psychological perspective conclusions are justified by effective, convincing arguments.

Also two type of argumentation were distinguished: rational and (di)a-rational argumentation. In case of rational argumentation the procedure of argumentation is of no importance. What only matters is that the premises are sufficient to accept the conclusion. In case of (di)a-rational argumentation the procedure is essential. Since the premises are not sufficient to accept the conclusion, why the conclusion was accepted can only be understood by analyzing the context in which the premises were adduced.

In a dialogical model of justification rational and (di)a-rational argumentation can and must be combined. DiaLaw is, to my knowledge, the only model that combines the two types of argumentation. Since rule based rationalism and rule skepticism as expressed by the American Realists are combined in one model, the theory underlying DiaLaw was called Rationalistic Realism.

3. Closing remarks

There will probably never be an ultimate model of legal justification. However, progress is possible, so it is worth to continue working on the development of still better models. Critical remarks about DiaLaw and evaluation of the model in practice will be just two sources of possible improvement.
For the future I mention two lines of further research. First, it is important to bridge the gap between theoretical models and models that are comprehensible for jurists. Tools need to be developed that represent the theories in a clear manner.

Second, in law school it is important to teach at an early stage legal skills in general and legal argumentation in particular. Although simple drill-and-practice programs are rather fruitful as computer assisted learning tools, the introduction of Intelligent Tutoring System in legal education is a goal worth striving for. For instance, debating techniques can be taught to a student by models such as DiaLaw [cf. Lodder, 1993; 1996a; Lodder & Verheij, 1998].

Taylor [1984, p. 74] once said:

"Something is just assumed to be wrong and other things not wrong, quite independently of any principles of right and wrong, and the whole discussion is aimed at finding some principle that will fit what is thus already assumed."

A few years ago I assumed that a dialogical model would be a good AI & Law model for legal justification. This thesis is the principle that I hope fits what I already assumed.

---

Appendix – Prolog code of DiaLaw

/*DiaLaw 2.0, 1998, Prolog 2 for Windows*/
clean :-
   retract(teller(N)),
   fail.
clean.

/*dialaw/0 is supposed to be true in the end*/
dialaw :-
clean,
   create_windows,
   first_move(CS, FirstSentence, DialogMoves),
   dialog(CS, FirstSentence, DialogMoves),
   writeln(move),
   write("In order to leave this DiaLaw-session, press any key: "),
   read(D),
   delete_windows.

/*first_move(CommitmentStore, FirstSentence, Dialogmoves)*/
The first move is different from the other moves, because a previous move lacks. The first act is always a claim of the first player conveniently called bert, his opponent is ernie/first_move([bert, Sentence]), Sentence, 
   [bert, (claim, Sentence), [0, dialaw]]) :-
   assert(teller(1)), teller(0),
   writeln(declare),
   write("on level 0, move 1 of the dialog game"),
   writeln(move),
   write("bert, act: claim", nl),
   write("sentence: ", Sentence),
   read(Sentence),
   writeln(dialog),
   write("bert: "),
   writeln([cs_bert]),
   write("ernie", " ", Sentence),
   writeln([cs_ernie]),
   write("ernie", " ", Sentence).

/*These are the conditions under which dialog/3 ends. Only if the first sentence (or in some case the denial) is accepted/withdrawn, there are no disputed sentences left (see def. 2). Quit is the other option.*/
dialog([S, [_ (accept, S), _|Rest]]).
dialog([S, [_ (withdraw, S), _|Rest]]).
dialog([S, [_ (quit, _, _|Rest)]].
dialog([S, [_ (accept, S), _|Rest]]).-- not_in([_, S], CS).

/*dialog(CommitmentStore, FirstSentence, DialogMoves)*/
This is the main part of the program. For each move the (recursive) clause
dialog/3 is called */
dialog(CS, FS, DM) :-
   move_input(CS, DM, NewPlayer, NewLevels, Move),
   valid_move(NewLevels, NewPlayer, CS, DM, ValidMove, Move),
   update(CS, FS, NewCS, ValidMove, NewPlayer),

/*move_input(CS, DialogMoves, NewPlayer, NewLevels, Move)*/
/*These are forced moves*/
/*Rule 6b*/
move_input(CS, [P, (accept, IL_claim(S)), [L, _]|_],
P, (L, il, claim(\$)), ![withdraw, S!]:-
otherplayer(P, Op),
only_inCS_other(CS, Op, S),
teller,
write_status([P, il, accept, S], ![L, !!!], P, (L, S)).
/*Rule 9a*/
move_input(CS, ![P, il, accept, A], ![L, _], ![P, (L, A)],
accept, reason(applicable(rule(A, B))), applies(rule(A, B))):-
incS(P, valid(rule(A, B))), CS,
only_inCS_other(CS, P, reason(applicable(rule(A, B))), applies(rule(A, B))),
teller,
write_status([P, il, accept, A], ![L, _], ![P, (L, A)]),
move_input(CS, ![P, il, accept, valid(rule(A, B))], ![L, _], ![P, ![L, valid(rule(A, B))]],
accept, reason(applicable(rule(A, B))), applies(rule(A, B))),
incS(P, P, A, CS),
only_inCS_other(CS, P, reason(applicable(rule(A, B))), applies(rule(A, B))),
teller,
write_status([P, il, accept, valid(rule(A, B))], ![L, !!!], P, ![L, valid(rule(A, B))]),
move_input(CS, ![P, claim, reason(applicable(rule(A, B))), applies(rule(A, B))],
![L, _], ![P, Op, ![L, reason(applicable(rule(A, B))), applies(rule(A, B))]],
accept, reason(applicable(rule(A, B))), applies(rule(A, B))),
teller,
write_status([P, claim, reason(applicable(rule(A, B))), applies(rule(A, B))], ![L, _], ![Op, ![L, reason(applicable(rule(A, B))), applies(rule(A, B))]]).
/*Rule 10a*/
move_input(CS, ![P, accept, applies(rule(A, B))], ![L, _], ![P, ![L, applies(rule(A, B))]],
accept, reason(A, B)),
only_inCS_other(CS, P, reason(A, B)),
teller,
write_status([P, accept, applies(rule(A, B))], ![L, _], ![P, ![L, applies(rule(A, B))]]).
/*Rule 10b*/
move_input(CS, ![P, accept, excluded(rule(A, B))], ![L, _], ![P, ![L, excluded(rule(A, B))]],
![withdraw, applies(rule(A, B))]),
otherplayer(P, Op),
only_inCS_other(CS, Op, applies(rule(A, B))),
teller,
write_status([P, accept, applies(rule(A, B))], ![L, _], ![P, ![L, applies(rule(A, B))]]).
/*Rule 14a*/
move_input(CS, ![P, accept, outweights(Pro, Con, S)], ![L, _], ![P, ![L, outweights(Pro, Con, S)]]),
![inCS(P, root S)], CS),
teller,
write_status([P, accept, outweights(Pro, Con, S)], ![L, _], ![P, ![L, outweights(Pro, Con, S)]]),
move_input(CS, ![P, [withdraw, not(S)]], ![L, _], ![P, ![L, not(S)]],
accept, S)),
inCS(P, outweights(Pro, Con, S)), CS),
teller,
write_status([P, [withdraw, not(S)]], ![L, _], ![P, ![L, outweights(Pro, Con, S)]]),
move_input(CS, ![P, accept, outweights(Pro, Con, S)], ![L, _], ![P, ![L, outweights(Pro, Con, S)]]),
teller,
write_status([P, accept, outweights(Pro, Con, S)], ![L, _], ![P, ![L, outweights(Pro, Con, S)]]).
/*Rule 14c*/
move_input(CS, ![P, accept, reason(A, not(S))], ![L, _], ![P, ![L, reason(A, not(S))]],
withdraw, outweights(Pro, Con, S))),
otherplayer(P, Op),
only_inCS adversely\(\text{other}(\text{CS, Op, outweighs(Pro, Con, S)}), \text{teller},\)\
\text{write_status}(\{\text{P, (accept, reason(R, not S))}, (L, _)\}, \text{P},\)\
\{L, reason(R, not S)\}).

/* general move input (voluntary, non-forced moves) */
move_input(\text{CS, DN, NewPlayer, NewLevels, Move}) :-
teller,
\text{next_move}(\text{DM, CS, NewPlayer, NewLevels}),
\text{write_status}(\text{DM, NewPlayer, NewLevels}),
\text{read_move}(\text{NewPlayer, Move}).

/* next_move(\text{DialogMoves, CommitmentStore, NewPlayer, NewLevels}) */
The new levels and who the new player is, depends on the previous move and previous player */
\text{next_move}(\{\text{P, (claim, B)}, (L, _)\}, \text{CS, NewPlayer, (L, B)}):-
\text{otherplayer}(\text{P, NewPlayer}),
\text{next_move}(\{\text{P, (question, _)}, (L, B)\}, \text{CS, NewPlayer, (LL, B)}):-
\text{otherplayer}(\text{P, NewPlayer}),
LL is L + 1.

/* Rule 5a */
\text{next_move}(\{\text{P, (withdraw, B)}, (_, _Rest)\}, \text{CS, P, (L, not(B))}) :-
\text{only_inCS other}(\text{CS, P, not(B)}),
\text{find_move}(\text{Rest, not(B)}, (L, _)).

/* Rule 5b */
\text{next_move}(\{\text{P, (accept, not(S))}, (_, _Rest)\}, \text{CS, P, (L, B)}):-
\text{find_move}(\text{Rest, S, (L, B)}),
\text{otherplayer}(\text{P, NewPlayer}).

/* Rule 8c */
\text{next_move}(\{\text{P, (withdraw, \texttt{il_claim(S)})}, (_, _Rest)\}, \text{CS, P, (L, S)}):-
\text{not_inCS}(_, not(\texttt{il_claim(S)})), \text{CS},
\text{find_move}(\text{Rest, S, (L, _)}).
\text{next_move}(\{\text{P, (withdraw, not(\texttt{il_claim(S))))), (_, _Rest)\}, \text{CS, Op, (L, S)} :-
\text{not_inCS}(_, \texttt{il_claim(S)}), \text{CS},
\text{otherplayer}(\text{P, Op}),
\text{find_move}(\text{Rest, S, (L, _)}).
\text{next_move}(\{\text{P, (accept, not(\texttt{il_claim(S)}))), (_, _Rest)\}, \text{CS, Op, (L, S)}):-
\text{find_move}(\text{Rest, S, (L, _)}).

/* Rule 16a */
\text{next_move}(\{\text{P, (withdraw, outweighs)\longrightarrow S1, (_, _Rest)\}, \text{CS, P, (L, reason(R, not(S))))}:-
\text{only_inCS other}(\text{P, reason(R, not(S))}, \text{CS}),
\text{find_move}(\text{Rest, reason(R, not(S))}, (L, _)).

/* Rule 16b */
\text{next_move}(\{\text{P, (accept, reason(R, not(S))))}, (_, _Rest)\}, \text{CS, P, (L, B)}):-
\text{find_move}(\text{Rest, reason(R, not(S))}, (L, outweighs\longrightarrow S1)),
\text{inCS}(\longrightarrow outweighs\longrightarrow S1), \text{CS},
\text{find_move}(\text{Rest, outweighs\longrightarrow S1), (L, B)}).
\text{next_move}(\{\text{P, (withdraw, reason(R, not(S))))}, (_, _Rest)\}, \text{CS, NewPlayer, (L, B)} :-
\text{inCS}(\longrightarrow outweighs\longrightarrow S1), \text{CS},
\text{find_move}(\text{Rest, outweighs\longrightarrow S1), (L, B)),
\text{otherplayer}(\text{P, NewPlayer}).

/* Rule 16c */
\text{next_move}(\{\text{P, (withdraw, reason(R, not(S))))}, (_, _Rest)\}, \text{CS, P, (L, B)}):-
\text{inCS}(\longrightarrow outweighs\longrightarrow S1), \text{CS},
\text{find_move}(\text{Rest, outweighs\longrightarrow S1), (L, B)}).

/* general accept\&\text{-}withdraw */
\text{next_move}(\{\text{P, (accept, S)}, (_, _Rest)\}, \text{CS, NewPlayer, (L, B)}):-
\text{find_move}(\text{Rest, S, (L, B)}),
\text{otherplayer}(\text{P, NewPlayer}),
\text{next_move}(\{\text{P, (withdraw, B)}, (_, _Rest)\}, \text{CS, P, NewLevels}).
find_move(Rest, B, NewLevels),

find_move([_, (claim, S), [L, B]]1, L, [L, B]), find_move([_|Rest, S, NL] :- find_move(Rest, S, NL).

/*write_status(DialogMoves, NewPlayer, NewLevels)*/
write_status([_, (A, S), _]|L, P, [L, B] :- write(text(status), nl).
        teller(U), write(U), write("The level is ", write(L, write("the move is ", write(N, write("", ").
        write[dialog], write(A), write("", ");
        write(S), nl, write(U), write("", "); write(player(P), write_levelmark(L).

writeplayer(Bert);:- write("Bert: ").
writeplayer(ernie);:- write("Ernie: ").

/*read_move(NewPlayer, Move)
move of the player is read*/
read_move(P, (A, S)) :-
    wText(text(move), [L, 0, 45, 630, 90], wText(move).
writeplayer(P), write("act: ");
read(A), write(" - sentence: ");
read(S), kIllpopup.

/*valid_move(NewLevels, NewPlayer, CommitmentStore, DialogMoves, ValidMove, Move)
checks the validity of a move after valid_move*/
valid_move(_, _, _, _, _, (quit, _), (quit, _)).

/* valid moves in case of accept*/
valid_move(_, NP, CS, _, (accept, S), (accept, S)) :- only_inCS_other(CS, NP, SI, not_LCSNP, not(S), CS).

/* valid moves in case of withdraw*/
valid_move(_, NP, CS, _, (withdraw, S), (withdraw, S)) :-
    otherplayer(NP, P, only_inCS_other(CS, P, S)).

/* valid moves in case of question*/
valid_move(NL, P, CS, [P, {withdraw, S, Levels}|Rest],
    [question, not(S)], [question, not(S)]) :-
otherplayer(P, Op, inCS!{Op, not S, CS}).


valid_move(NL, NP, CS, DM, ValidMove, (question, outweighs([_], [ ], [_])):-
  wininfo("Wrong move! Try again"),
  read_move(NP, Move),
  valid_move(NL, NP, CS, DM, ValidMove, Move).
valid_move(NL, NP, CS, ([P, (question, S1, L1)|Rest], ValidMove, (question, S1)):-
  wininfo("Wrong move! Try again"),
  read_move(NP, Move),
  valid_move(NL, NP, CS, ([P, (question, S1, L1)|Rest], ValidMove, Move).
valid_move(NL, NP, CS, ([(P, (question, S1, L1)|Rest], ValidMove, Move)_):-
  wininfo("Wrong move! Try again"),
  read_move(NP, Move),
  valid_move(NL, NP, CS, [(P, (question, S1, L1)|Rest], ValidMove, Move).
valid_move(NL, NP, CS, ValidMove, (question, reason(applicable(rule(A, B))), applies(rule(A, B)))):-
  inCS(NP, A, CS),
  inCS(NP, valid(rule(A, B)), CS),
  wininfo("Wrong move! Try again"),
  read_move(NP, Move),
  valid_move(NL, NP, CS, [(P, (question, S1, L1)|Rest], ValidMove, Move).
valid_move(_, _, CS, ([S2, (claim, S1, S2)|Rest], (question, S1), (question, S2)):-
  not_inCS(_, not(inCS(S1, S2))), CS),
  valid_move(_, _, CS, ([S2, (claim, S1, S2)|Rest], (question, S1), (question, S2)):-
  not_inCS(_, not(inCS(S1, S2))), CS).
valid_move(_, _, CS, ([S2, (claim, S1, S2)|Rest], (question, S1), (question, S2)):-
  not_inCS(_, not(inCS(S1, S2))), CS),
  otherplayer(NP, P),
  only_inCS_other(CS, P, outweighs([_, S1]), (question, S1), (question, S2)):-
  otherplayer(NP, P),
  previous_withdraw(NP, S, DM),
  wininfo("Wrong move! Try again"),
  read_move(NP, Move),
  valid_move(NL, NP, CS, ([P, (withdraw, S)|Rest], ValidMove, (claim, _)):-
  otherplayer(P, Op),
  inCS(Op, not not S), CS),
  wininfo("Wrong move! Try again"),
  read_move(NP, Move),
  valid_move(NL, NP, CS, ([P, (withdraw, S)|Rest], ValidMove, Move).
valid_move(NL, NP, CS, ([P, (withdraw, il_claim(S1)|Rest], ValidMove, (claim, _)):-
  wininfo("Wrong move! Try again"),
  read_move(NP, Move),
  valid_move(NL, NP, CS, ([P, (withdraw, il_claim(S1)|Rest], ValidMove, Move).
valid_move(NL, NP, CS, ([P, (withdraw, not(il_claim(S)))|Rest], Levels)|Rest),
  ValidMove, (claim, _)):-
  wininfo("Wrong move! Try again"),
  read_move(NP, Move),
  valid_move(NL, NP, CS, ([P, (withdraw, not(il_claim(S)))|Rest], Levels)|Rest),
  ValidMove, Move).
valid_move(NL, NP, CS, ([P, (accept, not(il_claim(S)))|Rest], Levels)|Rest),
  ValidMove, Move).
valid_move(NL, NP, CS, ([P, (accept, il_claim(S1))|Rest], Levels)|Rest),
  ValidMove, Move).
valid_move(NL, NP, CS, ([P, (accept, il_claim(S1))|Rest], Levels)|Rest),
  ValidMove, Move).
valid_move(NL, NP, CS, ([P, (accept, not(il_claim(S)))|Rest], Levels)|Rest),
  ValidMove, Move).
valid_move(Null, NP, CS, EM, ValidMove, (claim, applies(rule(A, B)))) :-
    wininfo("Wrong move! Try again"),
    read_move(NP, Move),
    valid_move(Null, NP, CS, EM, ValidMove, Move).

valid_move(_, applies(rule(A, B)), NP, CS, _,
    (claim, reason(applicable(rule(A, B)), applies(rule(A, B)))).
    (claim, not_inCS(_, reason(applicable(rule(A, B)), applies(rule(A, B))))),
    not_inCS(NP, not(A), CS),
    not_inCS(NP, not valido(rule(A, B))), CS).
valid_move(Null, NP, CS, EM, ValidMove,
    (claim, reason(applicable(rule(A, B)), applies(rule(A, B)))) :-
    wininfo("Wrong move! Try again"),
    read_move(NP, Move),
    valid_move(Null, NP, CS, EM, ValidMove, Move).

valid_move(_, CS, [_, (claim, B), _], (claim, il_claim(B)),
    (claim, il_claim(B)) :-
    B => il_claim()
valid_move(Null, NP, CS, EM, ValidMove, (claim, il_claim(B))) :-
    wininfo("Wrong move! Try again"),
    read_move(NP, Move),
    valid_move(Null, NP, CS, EM, ValidMove, Move).

valid_move(_, B, _CS, __claim, (claim, reason(A, B)),
    (claim, reason(A, B))) :-
    not_inCS(_, reason(A, B), CS).
valid_move(_, B, _CS, _claim, (claim, reason(A, not(B))),
    (claim, reason(A, not(B)))) :-
    not_inCS(_, reason(A, not(B)), CS).
valid_move(_, CS, [_, (claim, outweighs[_, _, B]], _]],
    (claim, reason(A, not(B))),
    (claim, reason(A, not(B)))) :-
    not_inCS(_, reason(A, not(B)), CS).
valid_move(Null, NP, CS, EM, ValidMove, (claim, reason[_, _]) :-
    wininfo("Wrong move! Try again"),
    read_move(NP, Move),
    valid_move(Null, NP, CS, EM, ValidMove, Move).

valid_move(_, B, _, CS, _claim, (claim, outweighs[Proset, Conset, B]],
    (claim, outweighs(Proset, Conset, B))) :-
    not_inCS(_, outweighs(Proset, Conset, B), CS),
    check_reasonsets(CS, Proset, Conset, B).

valid_move(_, _, [_, (claim, SI, _], (claim, not(SI))),
    (claim, not(SI))) :-
    S \= not(S).
valid_move(_, CS, [_, (question, _], _], (claim, S),
    (claim, S)) :-
    not_inCS(_, not(S), CS),
    not_inCS(_, not(S)), CS.
valid_move(_, CS, [_, (withdraw, _], _], (claim, S),
    (claim, S)) :-
    not_inCS(_, not(S), CS),
    not_inCS(_, not(S)), CS.
valid_move(_, CS, [_, (accept, _], _], (claim, S),
    (claim, S)) :-
    not_inCS(_, not(S), CS),
    not_inCS(_, not(S)), CS.
valid_move(Null, NP, CS, EM, ValidMove, _ :-
    wininfo("Wrong move! Try again"),
    read_move(NP, Move),
    valid_move(Null, NP, CS, EM, ValidMove, Move).

/*non-empty proset is not allowed*/
check_reasonsets(_, _, _), fail.
check_reasonsets(Null, Fvset, Conset, B) :-
    peel(Fvset, reason(R, not(B)), Rest),
    peel(Rest, Proset, Conset, B, Checkpro, Checkcon).

peel(Element, Rest, Proset, Conset, B) :-
    peel(Rest, Proset, Conset, B, Checkpro, Checkcon).

peel(_), fail.
Prolog code of DiaLaw

/*update_CS(DialogMoves, CS, NewCS, Move, NewPlayer)*/
update_CS(\, CS, [NP, SI|CS], (claim, S), NP):-
  write_CS([NP, S|CS]).
/*delete elements*/
update_CS(DM, CS, NewCS, (accept, S), NP):-
  bigupdate_CS(DM, CS, NewCS, (accept, S), NP),
  write_CS(NewCS).
update_CS(DM, CS, NewCS, (withdraw, S), NP):-
  bigupdate_CS(DM, CS, NewCS, (withdraw, S), NP),
  write_CS(NewCS).
update_CS(_, CS, _, _):-
  write_CS(CS).
/*bigupdate_CS(DialogMoves, CommitmentStore, NewCS, ValidMove, NewPlayer)*/
bigupdate_CS([], [claim, S], _, CS, [NP, SI|CS], (accept, S), NP),
bigupdate_CS([X|Res]CS, CS, NewCS, (accept, S), NP):-
  bigupdate_CS(Res, CS, NextCS, (accept, S), NP),
bigupdate_CS([], [claim, S], _, CS, NewCS, (withdraw, S), NP):-
  del([NP, S], CS, NewCS).
bigupdate_CS([X|Res]CS, CS, NewCS, (withdraw, S), NP):-
  bigupdate_CS(Res, CS, NextCS, (withdraw, S), NP),
/*realupdate_CS(DialogMoves, CommitmentStore, NewCS, ValidMove, NewPlayer),
for bigupdate here really the update of CS takes place*/
realupdate_CS([], [claim, not(S)], _, CS, CS, (withdraw, S), _, _):-
  inCS([p, claim, not(S)], CS),
  inCS([p, claim, S], CS, NewCS, _, _),
  inCS([p, S], CS),
  otherplayer(p, CS),
  not_inCS(op, S, CS),
  del([NP, S], CS, NewCS).realupdate_CS(_, CS, _, _).
write_CS(CS):-
  write_CS(CS, bert, CS1),
  telle(2),
  write("\n"),
  writeCS1(
    
    writeCS1 erneie,
    writeCS2),
  writeCS2.
write_CS([p, claim, not(S)], P, [S|Rest]], CS2):-
  write_CS(Rest, P, Rest1, CS2).
write_CS([p, claim, S], P, [S|Rest]], CS1, [S|Rest]]):-
  write_CS(Rest, P, CS1, [S|Rest]).
otherplayer(bert, ernie),
otherplayer(erneie, bert).
/*previous_withdraw(Player, S, DM)*/
previous_withdraw(\, _, _):-fai.
previous_withdraw(\, [I|I], _):-
previous_withdraw([\, S], S, I1),
previous_withdraw([\, S], S, I1).
only_inCS other(CS, P, S),
otherplayer(p, op),
inCS(op, S, CS).
not_inCS([P, S], CS).

inCS(_, []). :- fail.
inCS(Element, [Element|_]).
inCS(Element, [_|Rest]) :-
    inCS(Element, Rest).

not_inCS([_|]).
not_inCS(Element, [OtherElement|Rest]) :-
    Element \= OtherElement,
    not_inCS(Element, Rest).

del(X, [X|Tail], Tail).
del(X, [Y|Tail], [Y|Tail]) :-
    del(X, Tail, Tail).

create_windows :-
    wcreatetext(dialog, 1, 0, 135, 630, 205),
    wcreatetext(status, 1, 0, 0, 630, 45),
    wcreatetext(move, 1, 0, 45, 630, 90),
    wcreatetext(cs_bert, 1, 0, 340, 315, 100),
    wcreatetext(cs_ernie, 1, 316, 340, 315, 100),
    wsettext(dialog).

delete_windows :-
    wdeletetext(dialog),
    wdeletetext(status),
    wdeletetext(move),
    wdeletetext(cs_bert),
    wdeletetext(cs_ernie).

teller :-
    retract(teller(Umin)),
    U is Umin + 1,
    assert(teller(U)).

write_levelmark(0) :- !.
write_levelmark(L) :-
    write(" ");
    LL is L-1,
    write_levelmark(LL).

wrongmove(L) :-
    wcreatepopup(wrongmove, 310, 200, 200, 200),
    wcreatepopup(wrongmove, L, []).

killpopup :-
    wdeletetext(wrongmove).
killpopup.
Samenvatting

Dit proefschrift, ‘DiaLaw -- over juridische rechtvaardiging en dialoogspelen’, handelt over het formele en in Prolog geïmplementeerde dialoogmodel van juridische rechtvaardiging DiaLaw. Het maakt deel uit van het onderzoeksterrein Kunstmatige Intelligentie & Recht. Een eerste deel van dit onderzoeksgebied is het recht en juridisch rederen zo precies te beschrijven dat het mogelijk wordt om de resulterende modellen in de computer te implementeren. Een vraag die in dit verband rijst, is hoe het mogelijk is om het recht precies te reprentenen, als er niet eens consensus bestaat over wat recht precies is? Het helpen beantwoorden van deze vraag is een tweede deel van dit onderzoeksveld; geringelijk onderzoek zou omschrijven kunnen worden als formele of computacionele rechtsrhorie. Door precieze modellen (en computerprogramma’s) kan een bijdrage geholpen worden aan een beter inzicht in wat recht nu eigenlijk is. Het doel van dit proefschrift is een beter inzicht te verkrijgen in juridische rechtvaardiging.

Een inspirerend uitgangspunt van het huidige onderzoek is dat door overtuigende argumenten een schijnbaar verloren zaak gewonnen kan worden. Het doel van het onderzoek is het beantwoorden van de volgende vragen:

- Hoe kan bepaald worden of een juridische bewering gerechtvaardigd is en zijn er criteria waar een juridische bewering aan moet voldoen wil deze gerechtvaardigd zijn?
- Wat is een bevredigend model van juridische rechtvaardiging en is het mogelijk een dergelijk model te definiëren en implementeren?
- Hoe wordt in vergelijkbare modellen argumentatie gerepresenteerd?
- Wat is aanvaardbaar als een argument in juridische rechtvaardiging?

In hoofdstuk 2 wordt betoogd waarom juridische rechtvaardiging dialoogmatig gemodelleerd dient te worden. Eerst wordt het verschil tussen het product en het proces van rechtvaardiging uiteengezet. In de product-benadering is een bewering gerechtvaardigd als deze middels een geldige afleiding volgt uit gerechtvaardigde premissen. In de proces-benadering is een bewering gerechtvaardigd wanneer na een reeks van een of meer stappen de bewering gerechtvaardigd is volgens de regels van de procedure.

Vervolgens worden drie aspecten van het recht besproken die allen in opgeschouwd moeten worden genomen bij een model van juridische rechtvaardiging.

In de eerste plaats wordt de weerlegbaarheid van juridische conclusies besproken. Hiermee verband houdende bezwaren tegen een deductief model van rechtvaardiging zijn onder meer dat uitzonderingen expliciet in de voorwaarden moeten worden opgenomen en dat een deductief model niet goed overeenkomt met de wijze waarop juristen rederen. Op een bevredigende wijze modelleren van weerlegbaarheid vereist een niet-monotone logica.

In de tweede plaats wordt het open karakter van het recht besproken. Dit vereist dat het mogelijk moet zijn om wanneer een bewering gerechtvaardigd wordt, nieuwe regels en feiten toe te voegen. Het rechtvaardigingsmodel mag derhalve niet gesloten zijn. Hoewel het resultaat van toevoegingen van nieuwe regels en feiten als product gemodelleerd zou kunnen worden, is het toevoegen zelf een proces. Een model van rechtvaardiging moet derhalve procedureel zijn en toevoeging van informatie mogelijk maken.

In de derde plaats bestaat het zogeheten Münchhausen Trilemma, dat betrekking heeft op de problemen die spelen bij het fungeren van beweringen, ofwel het vinden van voldoende draagvlak, grond voor een conclusie. Wanneer een open model van
rechtsvaardiging gebruikt wordt, kan de rechtsvaardiging in beginsel eindeloos doorgaan. Immers, ieder argument of premisse aangevoerd om een bewering te rechtsvaardigen, is zelf een bewering die gerechtsvaardigd moet worden. Dit zogeheten oneindige regres kan op twee manieren gestopt worden. Door een cirkelredenering te gebruiken, of door de rechtsvaardiging op een willekeurig punt af te breken. Het is evident dat een cirkelredenering niet een bevredigende oplossing is. Het probleem van het afbreken op een willekeurig punt is dat dit het bestaan van juridische dogma’s veronderstelt, namelijk beweringen die geen rechtsvaardiging behoeven. Hoewel er wellicht dergelijke onwankelbare punten in het recht bestaan, gelet op de mogelijheid van het ter discussie stellen van een schijnbaar onwankelbaar punt niet afgesloten mag worden. Het Münchenhaus Trilemma behandelde derhalve drie onaangename situaties, oneindig regres, circulariteit en dogmatisering, waar een rechtsvaardiging toe kan leiden.

Om de besproken problematiek het hoofd te bieden, wordt een dialogisch model van rechtsvaardiging voorgesteld. Aangegeven wordt op welke wijze met de drie aspecten van het recht in een dialogisch model rekening kan worden gehouden. Een dialogisch model van rechtsvaardiging kan als volgt gekarakteriseerd worden:

- er zijn tenminste twee spelers;
- er zijn regels die het uitwisselen van beweringen tussen de spelers reguleren;
- beweringen zijn gerechtsvaardigd wanneer de spelers het eens zijn.

Tenslotte wordt gepoogd het dialogisch model van rechtsvaardiging te plaatsen binnen één van drie door Rawls gedefinieerde procedures, te weten de zuivere, de onvolmaakte en de volmaakte. Hoewel in eerder werk de procedure als zuiver omschreven is, is de conclusie nu dat het dialogische model onder geen van de genoemde procedures valt. Derhalve wordt een vierde procedure-type geïntroduceerd, die de juridische procedure wordt genoemd.

Conclusie van dit hoofdstuk is dat er geen oafhankelijk, materieel criterium bestaat met behulp waarvan kan worden vastgesteld of een bewering gerechtsvaardigd is. Daarom is het noodzakelijk een procedure te ontwikkelen waarbinnen beweringen gerechtsvaardigd kunnen worden. Deze procedure is vormgegeven als een dialoog, waarbinnen de spelers elkaar moeten proberen te overtuigen van de rechtvaardigheid van door hen aangevoerde beweringen. Of een bewering gerechtsvaardigd is, kan enkel worden vastgesteld in een dergelijke dialoog. De vraag of een binnen een procedure gerechtsvaardigde bewering ook rechtvaardig is, kan niet worden beantwoord. De uitdaging is om een procedure te ontwikkelen die garandeert dat iedere binnen de procedure gerechtsvaardigde bewering ook rechtvaardig is.

In hoofdstuk 3 wordt het dialogische model voor juridische rechtsvaardiging DiaLaw gedefinieerd. DiaLaw is in de eerste plaats een theoretisch, analytisch model. Met behulp van DiaLaw kan juridische rechtsvaardiging in het algemeen, en het proces van juridische rechtsvaardiging in het bijzonder, worden geanalyseerd. DiaLaw heeft ook een normatief aspect, omdat het richtlijnen geeft over hoe juridische rechtsvaardiging zou kunnen verlopen. DiaLaw is niet een empirisch model.

DiaLaw is een dialoogspel waarin door twee spelers afwisselend (argument)zetten kunnen worden gedaan. De bedoeling van het spel is het rechtsvaardigen van juridische beweringen door het aanvoeren van beweringen/argumenten en het bestrijden van beweringen/argumenten van de ander. De rechtsvaardiging heeft derhalve een sociaal karakter: het gaat erom de ander te overtuigen, van de (on)juistheid van beweringen.
De zetten bestaan uit een taalhandeling met een propositionele inhoud. De taalhandeling is één van de volgende vier soorten:
1. claim – het poneren van een bewering;
2. question – het vragen om rechtvaardiging van een bewering;
3. accept – het aanvaarden van een bewering;
4. withdraw – het intrekken van een eerder gesedane bewering.

Bij de taalhandelingen 2-4 is de inhoud van de bewering aan beperking onderhevig, aangezien deze taalhandelingen steeds ingaan op een eerdere bewering. Wanneer een speler een bewering claimt (taalhandeling 1), dan is de inhoud van de bewering in beginsel vrij invulbaar. De enige beperking is dan dat de dialoogregels het claimen van de bewering toestaan. Zo kan een speler bijvoorbeeld niet een bewering doen als hij zich eerder aan de ontkenning van de bewering gebonden heeft.

De argumentatie die gegeven wordt in de dialoog krijgt een inzichtelijke structuur doordat er verschillende niveaus zijn. Een dialoog begint op niveau 0. Het niveau wordt hoger wanneer een speler een bewering rechtvaardigd wil zien. De directe argumenten voor een propositie zijn daarom altijd één niveau hoger dan de bewering zelf. Deze structuur wordt visueel inzichtelijk gemaakt met behulp van dialoogbomen.

Of in een dialoog een bewering uiteindelijk rechtvaardigd wordt, is afhankelijk van de naar voren gebrachte beweringen/argumenten. In geval partijen er niet uitkomen, zullen ze naar een onafhankelijke derde, een rechter of arbiter, moeten stappen. In DiaLaw is de gang naar de rechter niet gemodelleerd.

Gebondenheid of commitment speelt een belangrijke rol in de dialoog. Commitment ontstaat in beginsel wanneer iemand een bewering claimt of accepteert, en eindigt op het moment dat hij een propositie intrekt. In een zogenaamde commitment store is opgeslagen aan welke proposities de spelers zijn gebonden. Een dialoog eindigt wanneer de eerste bewering wordt ingetrokken of geaccepteerd.

Als een zet niet kan, dan is er sprake van een ‘illegal move’ (vgl. het niet opheffen van schaak). Een speciale zet binnen de dialoog is die waarin een speler claimt dat de zet die de ander gedaan heeft niet mag. Dit is wat anders dan dat een zet niet kan. Een voorbeeld van een zet die niet mag is het claimen van een bewering die gebaseerd is op onrechtmatig verkregen bewijs. Of een zet niet mag wordt bepaald door de regels van het domein en dus niet door die van DiaLaw.

In 16 dialoogregels wordt beschreven wanneer een zet mogelijk is en wat de gevolgen van een zet zijn voor de commitment. Er is een onderscheid tussen algemene communicatieregels (regel 1-5) en speciale communicatieregels (regel 6-16). In de speciale communicatieregels worden de gevolgen geregeld van speciale (juridische) beweringen zoals redenen, het wegen van redenen, het toepassen van regels, en de eerdere genoemde bewering dat een bepaalde zet niet mag. Hoewel de spelers doorgaans beurtelings aan zet zijn, is in sommige gevallen na accept/withdraw dezelfde speler nogmals aan zet. Verder kan op grond van de speciale communicatieregels een speler gedwongen worden een bepaalde zet te doen. Wanneer een speler bijvoorbeeld accepteert dat de redenen voor een conclusie zwaarder wegen dan de redenen tegen de conclusie, dan is hij gedwongen de conclusie te accepteren.

DiaLaw is geïmplementeerd in Prolog (zie Appendix). In het Prolog-programma wordt bepaald welke speler aan zet is, of de door hem ingegeven zet geldig is (kan), wat de
gevolgen van een geldige zet zijn voor de commitment, en op welk niveau de dialoog zich afspeelt.

In hoofdstuk 4 wordt aan de hand van twee casus (Tyrell en Chabot) de werking van DiaLaw geïllustreerd. In de gegeven voorbeelddialogen zijn de definities en regels van hoofdstuk 3 toegepast en wordt geïllustreerd hoe juridische rechtvaardiging in DiaLaw gemodelleerd wordt.

In hoofdstuk 5 worden andere dialógische modellen met elkaar en met DiaLaw vergeleken. Meer uitgebreid aandacht wordt besteed aan drie modellen die het toevoegen van nieuwe informatie toestaan (open modellen), te weten Dialectics van Rescher, DC van MacKenzie en de Pleadings Game van Gorden. Centrale aandachtspunten zijn de gebruikte zetten en de afhandeling van commitment. De modellen kunnen als volgt gekarakteriseerd worden.

Een nadeel van formele dialoogmodellen is dat de resulterende dialoog vaak onnatuurlijk zijn. Dit geldt niet voor DC, maar een nadeel van dit systeem is dat onweerlegbare regels worden gebruikt. In Dialectics worden wel weergelagbare regels gebruikt. Een nadeel is dat de weergelaging enkel kan plaatsvinden op basis van specificiteit, en dat de regels niet ter discussie mogen worden gesteld. In de Pleadings Game wordt een rijke juridische taal gebruikt en kunnen regels ter discussie worden gesteld. Een nadeel is dat slechts echte argumenten op grond van de onderliggende logica mogen worden aangevoerd (vgl. Hoofdstuk 6) en dat de resulterende dialoog zeer gecompliceerd zijn. Deze en andere bezwaren van genoemde modellen gelden niet voor DiaLaw. Een nadeel van DiaLaw is dat een dialoog niet gestopt kan worden zonder deze te verliezen. Dit laatste is in de Pleadings Game fraai opgelost.

In hoofdstuk 6 wordt onderzocht wat nu preciezer als een argument gezien moet worden bij het rechtvaardigen van juridische beweringen. Daartoe wordt een onderscheid gemaakt tussen de logische en psychologische perspectief. In het logische perspectief zijn premissen voldoende om de conclusie te rechtvaardigen. Of premissen voldoende zijn, wordt bepaald door de regels van een logica. In geval van een Modus Ponens argument zijn bijvoorbeeld de premissen A en A → B, voldoende om de conclusie B te rechtvaardigen. In het logische perspectief spelen (afleid)regels een belangrijke rol.

In het psychologische perspectief wordt de conclusie gerechtvaardigd na daadwerkelijke aanvaarding, zonder dat de premissen voldoende behoeven te zijn in bovenbedoelde zin. In het psychologische perspectief gaat het erom dat de toehoorder bewogen wordt een bewering te accepteren.

Argumentatie bezien vanuit het logische perspectief wordt doorgaans rationele argumentatie genoemd; argumentatie bezien vanuit het psychologische perspectief kan als rationele argumentatie worden genoemd, en wordt in dit hoofdstuk dia-rationele argumentatie genoemd.

Iedere vorm van (dia-)rationele argumentatie kan door het toevoegen van een premisse rationeel gemaakt worden. Het verband dat op deze wijze aangebracht wordt tussen de beweringen en de conclusie kan bevredigend zijn voor hen die slechts gelegen in logisch geldige argumenten. Een dergelijke reconstructie toont echter niet de eigenlijke argumentatie, omdat in dat geval het begrip argument niet mede de retoriek omvat. Vanuit het oogpunt van de retoriek voegt de premisse niet echt iets toe aan de argumentatie, behalve een doorgaans onnatuurlijk verband tussen een verzameling beweringen en een conclusie.
In DiaLaw wordt regel gebaseerde rechtvaardiging (logische) gecombineerd met rechtvaardiging die niet op regels gebaseerd is (psychologische). Omdat in DiaLaw zowel regel-gebaseerd rationalisme, als het regel-scepticisme van onder meer de Amerikaanse Realisten wordt gemodellerd, wordt de theorie die aan DiaLaw ten grondslag ligt Rationalistic Realism genoemd.

Voor zover mij bekend is DiaLaw het eerste formele dialoogmodel waarin rationele en (di)a rationele argumentatie wordt gecombineerd, en is DiaLaw het enige model waarin een reden een onderliggende regel *kan* hebben, maar dit niet noodzakelijkerwijze *zo* is.

Toekomstige modellen van juridische rechtvaardiging zullen bestaande modellen, zoals DiaLaw, kunnen verbeteren. Het is daarbij echter van belang dat vastgehouden wordt aan het principe van de dialectiek. Daarnaast moet niet alleen aandacht worden geschonken aan logische geldigheid van argumenten, maar moet, ook in formele modellen, het retorische karakter van argumentatie tot uitdrukking komen. Slechts als aan deze voorwaarden is voldaan, kan argumentatie in het algemeen en juridische rechtvaardiging in het bijzonder goed worden gomodelleerd.
References


Gordon, T.F. (1996), Personal communication.


Lodder, A.R. (1993), Legal debate as an educational tool, in: *A glance at the conference Towards a global expert system in law* (abstracts), 1-3 Dec, Firenze.


Lorenzen, P. (1969), Normative logic and ethics, Bibliographischen Institut, Mannheim.


Author index

Aarnio .............................................. 19, 24, 119
Albert ............................................. 17, 95, 96
Alevin ............................................. 3, 128
Alexy .............................................. 19, 21, 89, 118
Ashley ............................................. 3

Barth .............................................. 19, 75
Bench-Capon ..................................... 4, 76, 79
Berman ............................................ 20
Branting ........................................... 3
Bratley ............................................ 104
Brewka .......................................... 75, 83, 84, 86-89, 93, 101-104, 114
Bachanan ......................................... 2

Chen ............................................. 76

De Vey Mestdagh ................................ 3
Denn Haan ........................................ 3
Dung ............................................. 106, 120
Dunne ............................................ 76

Farley ........................................... 7, 75, 104, 112, 113, 125, 127
Feteris ........................................... 7, 106
Finkelstein ....................................... 79
Freeman ......................................... 7, 75, 104, 112-114, 125, 127
Fuks .............................................. 79
Fuiler ............................................. 12

Gardner .......................................... 2, 3

Geffner .......................................... 93
Ginsberg ......................................... 13


Groendijk ........................................ 3
Grootendorst ................................... 29, 75, 79, 104-106, 125, 127

Habermas ....................................... 19
Hafner ............................................ 20
Hage ............................................. 3, 7, 20, 22-24, 27, 29, 47, 89, 126
Hamblin .......................................... 28, 32, 75, 79, 80-82

Hart .............................................. 4, 18
Headrick ........................................ 2
Herczog .......................................... 3, 7, 27, 29, 41, 47, 126
Hirsch-Ballin .................................... 14
Hutter ............................................ 3

Karacaplidis .................................... 104, 122
Kelsen .......................................... 18
Kloostermans ................................... 106
Kowalski ......................................... 7, 104
Krabbe .......................................... 19, 75
Kruiger .......................................... 79, 106
Leenes .......................................... 7, 22, 27, 29
Leijen .......................................... 14
Leng ............................................. 76
Lin ................................................ 120
Lorenz .......................................... 76, 118
Lorenzen ........................................ 19
Lowi ............................................. 7, 16, 75, 83, 89, 104, 106, 121
Lowes ........................................... 76, 79

MacKenzie ...................................... 5, 29, 32, 75-81, 87, 101-104, 114, 122, 125, 127
McEnery .......................................... 76, 79
Moles ............................................. 20
Moore ............................................ 79, 82
Nieuwenhuis .................................... 19
Nitta ............................................. 3, 7, 75, 76, 89, 104, 111, 118, 125, 127
Norman .......................................... 76, 121

Ohtake .......................................... 76, 89, 111
Olbrechts-Tyteca ................................ 76
Oskamp .......................................... 3

Pearl ............................................. 93
Peczenik ......................................... 19, 119, 121
Perelman ......................................... 19
Pilkington ....................................... 79
Plug ............................................. 106
Pollock .......................................... 9, 12, 106
Poulin .......................................... 76, 104
Pound .......................................... 76, 73
<table>
<thead>
<tr>
<th>Author</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prakken</td>
<td>3, 7, 13, 76, 89, 104, 106, 121</td>
</tr>
<tr>
<td>Rawls</td>
<td>8, 22</td>
</tr>
<tr>
<td>Reiter</td>
<td>13</td>
</tr>
<tr>
<td>Rissland</td>
<td>3</td>
</tr>
<tr>
<td>Rödig</td>
<td>16</td>
</tr>
<tr>
<td>Sartor</td>
<td>3, 7, 76, 104, 106, 121</td>
</tr>
<tr>
<td>Schooten</td>
<td>4, 14</td>
</tr>
<tr>
<td>Schwemmer</td>
<td>19</td>
</tr>
<tr>
<td>Searle</td>
<td>29</td>
</tr>
<tr>
<td>Sergot</td>
<td>2</td>
</tr>
<tr>
<td>Shibasaki</td>
<td>7, 76, 111</td>
</tr>
<tr>
<td>Shoham</td>
<td>120</td>
</tr>
<tr>
<td>Simari</td>
<td>106</td>
</tr>
<tr>
<td>Skalak</td>
<td>3</td>
</tr>
<tr>
<td>Smith</td>
<td>3, 18</td>
</tr>
<tr>
<td>Span</td>
<td>3, 7, 27, 29, 89</td>
</tr>
<tr>
<td>Stevenson</td>
<td>9, 118</td>
</tr>
<tr>
<td>St-Vincent</td>
<td>76, 104</td>
</tr>
<tr>
<td>Susskind</td>
<td>2</td>
</tr>
<tr>
<td>Taylor</td>
<td>128</td>
</tr>
<tr>
<td>Toni</td>
<td>7, 104</td>
</tr>
<tr>
<td>Toulmin</td>
<td>89, 95, '100, 112</td>
</tr>
<tr>
<td>Trager</td>
<td>3</td>
</tr>
<tr>
<td>Unger</td>
<td>4</td>
</tr>
<tr>
<td>Valente</td>
<td>3</td>
</tr>
<tr>
<td>Van Dalen</td>
<td>34, 118</td>
</tr>
<tr>
<td>Van den Herik</td>
<td>1</td>
</tr>
<tr>
<td>Van Eemeren</td>
<td>29, 75, 79, 104-106, 125, 127</td>
</tr>
<tr>
<td>Van Kralingen</td>
<td>3</td>
</tr>
<tr>
<td>Verheij</td>
<td>3, 20, 47, 106, 126, 128</td>
</tr>
<tr>
<td>Visser</td>
<td>3</td>
</tr>
<tr>
<td>Voermans</td>
<td>3</td>
</tr>
<tr>
<td>Vossos</td>
<td>3</td>
</tr>
<tr>
<td>Vreeswijk</td>
<td>22, 75, 76, 83, 104, 106, 107, 109-111, 118, 120, 125, 127</td>
</tr>
<tr>
<td>Wahlgren</td>
<td>3</td>
</tr>
<tr>
<td>Walker</td>
<td>3</td>
</tr>
<tr>
<td>Walser</td>
<td>2</td>
</tr>
<tr>
<td>Walton</td>
<td>75, 83</td>
</tr>
<tr>
<td>Wellman</td>
<td>9</td>
</tr>
<tr>
<td>Witteveen</td>
<td>20</td>
</tr>
<tr>
<td>Wong</td>
<td>76, 89, 111</td>
</tr>
<tr>
<td>Woods</td>
<td>83</td>
</tr>
<tr>
<td>Zeleznikov</td>
<td>3</td>
</tr>
</tbody>
</table>
# Subject index

<table>
<thead>
<tr>
<th>Term</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>agents</td>
<td>76</td>
</tr>
<tr>
<td>AI &amp; Law</td>
<td>2, 3, 76, 122, 128</td>
</tr>
<tr>
<td>altering dialog rules</td>
<td>21</td>
</tr>
<tr>
<td>applicable</td>
<td>15f., 23, 53f., 69f., 90f., 119f.</td>
</tr>
<tr>
<td>arbiter</td>
<td>29</td>
</tr>
<tr>
<td>argument</td>
<td>10, 11, 16</td>
</tr>
<tr>
<td>defeasible</td>
<td>89, 107, 109, 118</td>
</tr>
<tr>
<td>open</td>
<td>110</td>
</tr>
<tr>
<td>valid</td>
<td>10, 12, 28, 118, 119, 121, 127</td>
</tr>
<tr>
<td>argumentation</td>
<td>5, 117f.</td>
</tr>
<tr>
<td>(di)ra-tional</td>
<td>9</td>
</tr>
<tr>
<td>dynamic</td>
<td>104, 106, 120</td>
</tr>
<tr>
<td>formal</td>
<td>119f.</td>
</tr>
<tr>
<td>rational</td>
<td>8, 9, 15</td>
</tr>
<tr>
<td>static</td>
<td>5</td>
</tr>
<tr>
<td>argument process</td>
<td>18, 39, 41, 99, 110</td>
</tr>
<tr>
<td>arguments against</td>
<td>2, 3</td>
</tr>
<tr>
<td>artificial intelligence</td>
<td>1</td>
</tr>
<tr>
<td>assertion</td>
<td>1, 76, 106</td>
</tr>
<tr>
<td>cautious</td>
<td>86, 93</td>
</tr>
<tr>
<td>defensibility</td>
<td>7, 12, 18, 20, 84</td>
</tr>
<tr>
<td>defeasible</td>
<td>106</td>
</tr>
<tr>
<td>defeat</td>
<td>12, 20, 78, 193, 107f., 114, 127</td>
</tr>
<tr>
<td>defendant</td>
<td>78, 89-95, 111, 112</td>
</tr>
<tr>
<td>determiner</td>
<td>83, 86</td>
</tr>
<tr>
<td>dialectics</td>
<td>75f., 83, 106, 111, 115, 125f.</td>
</tr>
<tr>
<td>computational</td>
<td>83</td>
</tr>
<tr>
<td>formal</td>
<td>83, 88</td>
</tr>
<tr>
<td>dialog</td>
<td>19</td>
</tr>
<tr>
<td>one-person</td>
<td>19, 28</td>
</tr>
<tr>
<td>two-person</td>
<td>3, 5, 19-21, 24, 27, 29, 75, 76, 79, 81, 83, 88, 125, 126</td>
</tr>
<tr>
<td>dialog game</td>
<td>5, 33, 37, 38, 40, 65, 67, 126</td>
</tr>
<tr>
<td>dialog approach</td>
<td>7, 122</td>
</tr>
<tr>
<td>disagreement</td>
<td>23, 29, 91</td>
</tr>
<tr>
<td>discourse rules</td>
<td>21</td>
</tr>
<tr>
<td>Dutch Supreme Court decisions</td>
<td>1954, Nov. 9, NJ 1955, 55.</td>
</tr>
<tr>
<td>1990, June 1, NJ 1991, 720</td>
<td></td>
</tr>
<tr>
<td>1994, June 21, NJ 1994, 656</td>
<td></td>
</tr>
<tr>
<td>1997, Feb. 21, RVdW 1997, 54C.</td>
<td></td>
</tr>
<tr>
<td>Erlangen School</td>
<td>19</td>
</tr>
<tr>
<td>euthanasia</td>
<td>5, 62, 68, 69, 72</td>
</tr>
<tr>
<td>evidence</td>
<td>8, 49, 63-66, 68, 78, 80, 83, 90, 93, 99, 102, 103, 107, 113, 115</td>
</tr>
<tr>
<td>excluded</td>
<td>16, 47, 48, 50, 54, 76-72, 93</td>
</tr>
<tr>
<td>Term</td>
<td>Page Numbers</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>extensional interpretation</td>
<td>93</td>
</tr>
<tr>
<td>first move</td>
<td>30-32, 35-38, 40, 59, 60, 69, 78, 83, 85, 86, 90, 107</td>
</tr>
<tr>
<td>fixed set</td>
<td>20, 75</td>
</tr>
<tr>
<td>force majeure</td>
<td>68, 71-73, 121</td>
</tr>
<tr>
<td>forced</td>
<td></td>
</tr>
<tr>
<td>-acceptance</td>
<td>56, 71, 123</td>
</tr>
<tr>
<td>-commitment</td>
<td>27, 32, 41, 50</td>
</tr>
<tr>
<td>-withdrawal</td>
<td>52, 58, 67, 123</td>
</tr>
<tr>
<td>formal grounds</td>
<td>86</td>
</tr>
<tr>
<td>fundamental move</td>
<td>83-85</td>
</tr>
<tr>
<td>game record</td>
<td>80, 86, 92</td>
</tr>
<tr>
<td>grounds</td>
<td>8, 9, 24, 68</td>
</tr>
<tr>
<td>procedural</td>
<td>110</td>
</tr>
<tr>
<td>substantial</td>
<td>110</td>
</tr>
<tr>
<td>Grundnorm (Kelsen)</td>
<td>17</td>
</tr>
<tr>
<td>ICAIL</td>
<td>2, 89</td>
</tr>
<tr>
<td>il claim</td>
<td>47, 49-53</td>
</tr>
<tr>
<td>illegally obtained evidence</td>
<td>5, 49, 64, 65</td>
</tr>
<tr>
<td>illocutionary act</td>
<td>29, 30, 32, 34, 35, 38, 40, 59-61, 63, 79</td>
</tr>
<tr>
<td>independent criterion</td>
<td>19, 22-24, 29, 125</td>
</tr>
<tr>
<td>information retrieval</td>
<td>2</td>
</tr>
<tr>
<td>information technology</td>
<td>1</td>
</tr>
<tr>
<td>interaction</td>
<td>38, 39, 76</td>
</tr>
<tr>
<td>interior dialog</td>
<td>105</td>
</tr>
<tr>
<td>interpretation</td>
<td>15</td>
</tr>
<tr>
<td>issue</td>
<td>8, 14, 29, 90, 97, 98, 105, 106, 111, 112</td>
</tr>
<tr>
<td>judge</td>
<td>4, 9, 20, 23, 28, 29, 49</td>
</tr>
<tr>
<td>human</td>
<td>1</td>
</tr>
<tr>
<td>judicial decision</td>
<td>9, 106</td>
</tr>
<tr>
<td>JURIX</td>
<td>2, 7</td>
</tr>
<tr>
<td>justification</td>
<td></td>
</tr>
<tr>
<td>-of dialog rules</td>
<td>21</td>
</tr>
<tr>
<td>depth of the</td>
<td>69</td>
</tr>
<tr>
<td>procedure of</td>
<td>21, 23, 24</td>
</tr>
<tr>
<td>process of</td>
<td>5, 7, 8, 9, 21, 126</td>
</tr>
<tr>
<td>product of</td>
<td>8, 9, 15, 19, 22</td>
</tr>
<tr>
<td>justified</td>
<td>12</td>
</tr>
<tr>
<td>-conclusion</td>
<td></td>
</tr>
<tr>
<td>-premise</td>
<td>10, 11, 16, 18</td>
</tr>
<tr>
<td>-statement</td>
<td>23, 24, 28, 115, 125</td>
</tr>
<tr>
<td>layered dialog</td>
<td>39, 108, 126</td>
</tr>
<tr>
<td>layout of the moves</td>
<td>63</td>
</tr>
<tr>
<td>legal doctrine</td>
<td>68</td>
</tr>
<tr>
<td>legal education</td>
<td>2, 128</td>
</tr>
<tr>
<td>legal justification</td>
<td>3-5, 7, 11, 16, 18, 20, 22-25, 27, 28, 63, 73, 76, 83, 115, 119, 122, 124-128</td>
</tr>
<tr>
<td>defeasible nature of</td>
<td>7, 11, 20, 126</td>
</tr>
<tr>
<td>legal reasoning</td>
<td>2, 3, 7, 11, 12, 18, 20, 76, 98, 100, 106</td>
</tr>
<tr>
<td>level</td>
<td></td>
</tr>
<tr>
<td>higher</td>
<td>35, 41, 43, 108, 109</td>
</tr>
<tr>
<td>lower</td>
<td>39, 41, 108, 109</td>
</tr>
<tr>
<td>lecution</td>
<td>39, 76, 79, 80</td>
</tr>
<tr>
<td>logic</td>
<td></td>
</tr>
<tr>
<td>nonmonotonic</td>
<td>10, 13, 106</td>
</tr>
<tr>
<td>main claim</td>
<td>90, 92</td>
</tr>
<tr>
<td>main thesis</td>
<td>107, 108, 110</td>
</tr>
<tr>
<td>marihuana</td>
<td>64-66</td>
</tr>
<tr>
<td>material grounds</td>
<td>86</td>
</tr>
<tr>
<td>material implication</td>
<td>10, 80, 82, 83, 93, 101, 104, 122</td>
</tr>
<tr>
<td>model</td>
<td></td>
</tr>
<tr>
<td>AI</td>
<td>3</td>
</tr>
<tr>
<td>AI &amp; Law</td>
<td>76, 122, 128</td>
</tr>
<tr>
<td>analytical</td>
<td>4</td>
</tr>
<tr>
<td>deductive</td>
<td>7, 10-12, 126</td>
</tr>
<tr>
<td>dialectical</td>
<td>5, 89, 104, 106</td>
</tr>
<tr>
<td>dialogical</td>
<td>3, 5, 7, 8, 18-20, 22, 24, 25, 27, 75, 89, 104, 117, 122, 124-128</td>
</tr>
<tr>
<td>empirical</td>
<td>4</td>
</tr>
<tr>
<td>formal</td>
<td>1</td>
</tr>
<tr>
<td>normative</td>
<td>4, 89</td>
</tr>
<tr>
<td>open</td>
<td>75, 125, 127</td>
</tr>
<tr>
<td>procedural</td>
<td>8, 16, 18, 19, 21</td>
</tr>
<tr>
<td>subsumption</td>
<td>11-16, 20</td>
</tr>
<tr>
<td>model of justification</td>
<td>5, 10, 22, 126</td>
</tr>
<tr>
<td>monological proof</td>
<td>7</td>
</tr>
<tr>
<td>Münchhausen Trilemma</td>
<td>8, 16f.</td>
</tr>
<tr>
<td>Topic</td>
<td>References</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Nixon paradox</td>
<td>88</td>
</tr>
<tr>
<td>non-deterministic</td>
<td>20</td>
</tr>
<tr>
<td>Obligation Game</td>
<td>76</td>
</tr>
<tr>
<td>open nature of law</td>
<td>7, 14, 20, 126</td>
</tr>
<tr>
<td>opportunity principle</td>
<td>68</td>
</tr>
<tr>
<td>ordering</td>
<td>93, 107, 111, 112</td>
</tr>
<tr>
<td>basic</td>
<td>107, 108</td>
</tr>
<tr>
<td>special</td>
<td>107</td>
</tr>
<tr>
<td>perspectives, logical vs. psychological</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5, 117-119, 127</td>
</tr>
<tr>
<td>Pleadings Game</td>
<td>3, 5, 75-77, 89-102, 104, 107, 111, 114, 115, 122, 125, 127</td>
</tr>
<tr>
<td>principles</td>
<td>48, 89, 128</td>
</tr>
<tr>
<td>probable cause</td>
<td>64, 66</td>
</tr>
<tr>
<td>procedure</td>
<td></td>
</tr>
<tr>
<td>‘legal’</td>
<td>8, 22, 24, 31</td>
</tr>
<tr>
<td>Rawls’ imperfect</td>
<td>22, 24</td>
</tr>
<tr>
<td>Rawls’ perfect</td>
<td>22</td>
</tr>
<tr>
<td>Rawls’ pure</td>
<td>8, 22, 23</td>
</tr>
<tr>
<td>rhetorical</td>
<td>20</td>
</tr>
<tr>
<td>propositional content</td>
<td>29, 30, 34, 35, 38, 43, 59, 61, 79</td>
</tr>
<tr>
<td>Protagoras</td>
<td>19</td>
</tr>
<tr>
<td>question begging</td>
<td>80</td>
</tr>
<tr>
<td>rationale</td>
<td>41, 76</td>
</tr>
<tr>
<td>reaction-based game</td>
<td>44, 126</td>
</tr>
<tr>
<td>reason</td>
<td></td>
</tr>
<tr>
<td>-con</td>
<td>32, 47-49, 55-58, 67, 72, 118, 123</td>
</tr>
<tr>
<td>-pro</td>
<td>7, 19-21, 47-49, 55, 56, 63, 71, 72, 82, 118, 121, 123</td>
</tr>
<tr>
<td>Reason-Based Logic</td>
<td>20, 27, 47, 48, 126</td>
</tr>
<tr>
<td>reasoning scheme</td>
<td>10, 19</td>
</tr>
<tr>
<td>rebutter</td>
<td>12</td>
</tr>
<tr>
<td>relevancy</td>
<td>99, 110</td>
</tr>
<tr>
<td>resolution demand</td>
<td>81, 302</td>
</tr>
<tr>
<td>retractions</td>
<td>101</td>
</tr>
<tr>
<td>rules</td>
<td></td>
</tr>
<tr>
<td>exceptions to</td>
<td>12</td>
</tr>
<tr>
<td>conflict of</td>
<td>13, 93, 160</td>
</tr>
<tr>
<td>purpose of</td>
<td>48</td>
</tr>
<tr>
<td>rule of recognition (Hart)</td>
<td>17</td>
</tr>
<tr>
<td>rules for communication</td>
<td></td>
</tr>
<tr>
<td>general</td>
<td>27, 50</td>
</tr>
<tr>
<td>special</td>
<td>27, 47, 50</td>
</tr>
<tr>
<td>rules of the procedure</td>
<td>9</td>
</tr>
<tr>
<td>sentence</td>
<td></td>
</tr>
<tr>
<td>disputed</td>
<td>36, 41, 64, 67, 92</td>
</tr>
<tr>
<td>first</td>
<td>39, 60</td>
</tr>
<tr>
<td>open</td>
<td>36, 41, 52, 54, 55, 60, 67</td>
</tr>
<tr>
<td>questioned</td>
<td>39, 44</td>
</tr>
<tr>
<td>silent implies consent</td>
<td>99, 101</td>
</tr>
<tr>
<td>skeptical consequence</td>
<td>86</td>
</tr>
<tr>
<td>special language elements</td>
<td>27, 28, 39, 47-50, 126</td>
</tr>
<tr>
<td>state of affairs</td>
<td>34, 47</td>
</tr>
<tr>
<td>statement</td>
<td></td>
</tr>
<tr>
<td>challenged</td>
<td>81, 122</td>
</tr>
<tr>
<td>- is justified</td>
<td>4, 8, 9, 20, 22-24, 28, 125</td>
</tr>
<tr>
<td>questioned</td>
<td>28</td>
</tr>
<tr>
<td>strategy</td>
<td>69</td>
</tr>
<tr>
<td>support relation</td>
<td>47</td>
</tr>
<tr>
<td>sufficient to accept</td>
<td>120-125, 127</td>
</tr>
<tr>
<td>symmetric game</td>
<td>88</td>
</tr>
<tr>
<td>tautology</td>
<td>76</td>
</tr>
<tr>
<td>the game H</td>
<td>79</td>
</tr>
<tr>
<td>Tyrell</td>
<td>61, 63-67, 76, 78, 127</td>
</tr>
<tr>
<td>undercutter</td>
<td>12</td>
</tr>
<tr>
<td>vagueness of legal language</td>
<td>14</td>
</tr>
<tr>
<td>valid inference</td>
<td>8, 120</td>
</tr>
<tr>
<td>validity of a rule</td>
<td>65</td>
</tr>
<tr>
<td>viewpoint</td>
<td>111</td>
</tr>
<tr>
<td>weighing</td>
<td>47, 48, 54, 55, 110</td>
</tr>
<tr>
<td>trivial</td>
<td>47</td>
</tr>
</tbody>
</table>
Curriculum vitae


In 1992 he started working as a Ph.D-student (AIO) at the Universiteit Maastricht, department of Metajuridica. Beside being a Ph.D, he was appointed as the coordinator of computer assisted learning (1993-1996) and as lecturer (1996-) at the Universiteit Maastricht. In 1998 he started working at the Computer/Law Institute of the Vrije Universiteit Amsterdam.

He has been a member of the ‘AIO-commissie’ (1992-1997). He co-organized the Dutch workshop on dialogical argumentation DIA '94, co-organized the eighth international conference on legal knowledge-based systems JURIX '95, co-organized the international workshop on argumentation MAD '96, and this year co-organizes ‘Views on legal argumentation, an international workshop’ and JURIX '98.

Currently he is working at both the Universiteit Maastricht and the Vrije Universiteit Amsterdam.
Contact information

Arno R. Lodder

Universiteit Maastricht
Faculty of Law
Department of Metajuridica
P.O. Box 616
6200 MD Maastricht
The Netherlands
+31-(0)43-3883005
arno.lodder@metajur.unimaas.nl
http://www.metajur.unimaas.nl/~arno

Vrije Universiteit Amsterdam
Faculty of Law
Computer/Law Institute
De Boelelaan 1105
1081 HV Amsterdam
The Netherlands
+31-(0)20-4446215
lodder@rechten.vu.nl
http://www.rechten.vu.nl/~lodder