Bachelor thesis

Artificial Intelligence

Radboud University

Solving legal cases with and without modal logic

Author:
Bibian Bennink
S1002645

First supervisor:
dr. E.M.G.M. (Engelbert) HUBBERS
Digital Security,
Institute for Computing and Information Sciences

Second supervisor:
dr. F. (Freek) WIEDIJK
Software Science,
Institute for Computing and Information Sciences

July 10, 2020

Abstract

The research presented in this thesis elaborates on whether modal logic has an advantage over models without modal logic in constructing models for legal cases. There are several kinds of logics that could be used to construct such a model. Within this thesis for the model with modal logic: Standard Deontic Logic (SDL) together with Propositional Deontic Logic (PD_eL) is used. For the model without modal logic: a reduction graph representation is used.

This thesis starts off with theoretical background on both SDL and PD_eL, and the reduction graph representation, where the background for the models used in this thesis is given. Secondly, the models are constructed for two legal cases, the Bourhill v Young case and HI's case. Finally, a comparison is made between the logical models to conclude whether, in the cases used in this thesis, deontic logic is an advantage. From the comparison it can be concluded that, for the cases used in this thesis, using deontic logic is not necessarily an advantage over using the reduction graph representation. This is due to the fact that deontic logic is relatively complex and does not yet deal well with cases that need a subjective approach.

Contents

Chapter 1

Introduction

Artificial intelligence is more and more common in the field of law, but as stated by Surden [27]: "AI is not magic". Currently the two broad categories with the most success are Machine Learning, and Rules, Logic and Knowledge representation [27, 9, 12]. Machine Learning approaches are currently mostly used in law since they try to mimic the way humans learn [27]. There are however several drawbacks of using Machine Learning, for example overfitting or the need of much data. A system of Knowledge representation, Logic and Rules on the other hand does not learn from data, also known as a bottom-up approach, but uses a top-down approach. This means that before the system is used, all possibilities need to be in the system already. The relationship between logic and law has been troublesome over the past century [12]. Law is one of the disciplines where it is difficult to use AI in, since law is not only about strict rules but also about "common sense". When looking at what kind of logic is used several papers can be found that discuss the use of modal logic, specifically deontic logic [14, 31]. Modal logic is an umbrella term that covers the logic of different sorts of modalities. A more elaborate explanation is given in chapter 2. Branting on the other hand describes how reduction graphs can be used to represent legal cases [7]. A reduction graph is a representation of the judgement of a certain predicate. A more elaborate explanation is given in chapter 2.

This thesis will look into the difference between using modal logic and using reduction graphs, which do not use modal logic. The research question for this thesis is:

Is using modal logic, specifically deontic logic, an advantage in logical models for solving legal problems?

In this thesis only two logical models will be compared on two different cases, therefore it cannot be generally concluded whether using modal logic is an advantage over not using modal logic. However, it is a start.

In chapter 2 both deontic logic and reduction graphs are further elaborated on. In chapter 3 the legal models will be shown. In chapter 4 the models will be compared. Lastly in chapter 5 a conclusion will be drawn.

Chapter 2

Theoretical background

2.1 Preliminaries

Before reading this thesis several definitions need to be clarified.

- **Logical model**: A logical model is a model of the specific case using expressions of a specific kind of logic. This can also be called a logical framework.

- **Norm**: A norm is a standard for evaluating or making judgements about behaviour or outcomes [26].

- **Normative judgement**: A normative judgement is a judgement about a normative situation such as rights and duties [28].

- **Legal authorities**: Legal authorities is any provision of law or regulation that carries the force of law [17]. Examples of legal authorities are statutes, rules and regulations, court rulings etc.

- **Warrant**: A warrant is a general or hypothetical statement that can act as a bridge. They can be very simple, but often of the form: "If A, then B" [25].

Within this thesis the basic principles and terminology of logic is used and therefore not elaborated on. This knowledge is based on the syllabus from the course Formal Reasoning at the Radboud University (IPK001), the syllabus can be found at: `https://www.cs.ru.nl/~freek/courses/fr-2017/public/fr.pdf`.

Before the logical models can be made or elaborated on, theoretical background on modal, particularly deontic logic, and reduction graphs is needed.

2.2 Modal logic: deontic logic

This section is split up in three subsections. In the section 2.2.1 modal logic in general is elaborated on, in section 2.2.2 deontic logic in specific is elaborated on. In section 2.2.3 two forms of deontic logic that will be used in this thesis are further elaborated on.

2.2.1 Modal logic

Striclty speaking, modal logic is the study of the deductive behaviour of the the expressions "it is necessary that" and "it is possible that" [11]. Deductive reasoning is reasoning from general to a specific case [13]. For example the reasoning "Alice has a lot of dresses" is deducted from: "Every woman has a lot of dresses" and "Alice is a woman". Broadly speaking it can be viewed as the logic of different sorts of modalities (or modes of truth) [4]. It is often used to describe a family of logics, among these logics are alethic logic, temporal logic, deontic logic and doxastic logic. There are essential similarities between these logics, but they also have different characteristics [30]. The term "modal logic" is reserved for the logic of the alethic modalities, which studies the expressions "necessarily" and "possibly" using the operator $\Box$ for "It is necessary that ..." and the operator $\Diamond$ for "It is possible that ..." [19]. There are five statuses in alethic modal logic:

1. It is necessary that ...
2. It is possible that ...
3. It is impossible that...
4. It is non-necessary that ...
5. It is contingent that ...

The fourth status is the negation of the first status, and the third status is the negation of the second status. As shown in figure 2.1, these statuses can also be depicted in a modal square of opposition. This visually shows how the statuses are related to one another. The modal square of opposition is based upon Aristotle's notion of contrary (two propositions cannot both be true, but they can both be false) and contradictory (the truth of one proposition implies the falsity of the other proposition) [21, 3]. A proposition can only be in one state at the same time, which is also intuitive for the statuses.

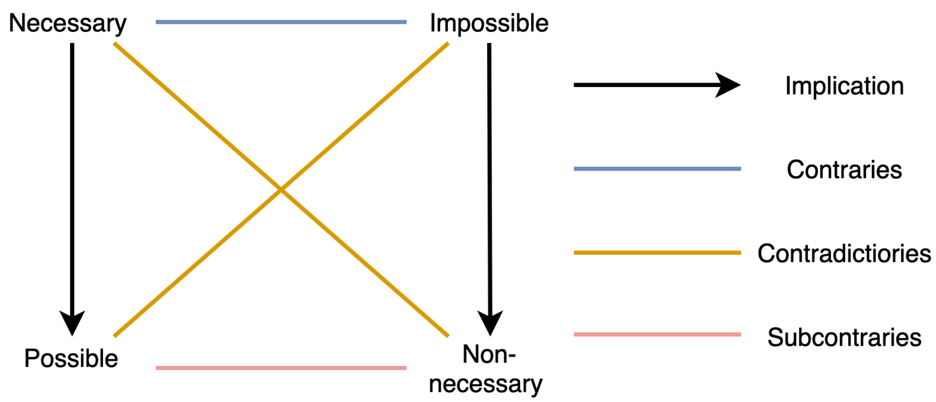

Figure 2.1: Modal square of opposition [19])

All of these statuses can be defined using the first status and operator ($\Box$), the negation operator ($\neg$) and the and-operator ($\wedge$). r is used as proposition:

1. $\Box r$
2. $\neg\Box\neg r$
3. $\neg\Box r$
4. $\Box\neg r$
5. $\neg\Box r \wedge \neg\Box\neg r$

Modal logic is most of the time constructed from a relatively weak logic: **K** [11], which is named after Saul Kripke. To get the logic **K**, the following need to be added to propositional logic, where T and Q are formulas:

1. $\Box$: The operator for "It is necessary that ..."
2. **Necessity rule:** If T is a theorem of **K** then $\Box T$ is too.
3. **Distribution Axiom**: $\Box(T \to Q) \to (\Box T \to \Box Q)$

The necessity rule is saying that the rules of logic are necessary and true in all possible worlds. The distribution axiom states that when an implication is necessary, the participants of that implication statement are necessary when used in another implication statement. The Necessity is distributed over T and Q in our case.

The logic **K** is the basis for most modal logics [19]. Often systems are added to **K** to make them more suitable for a certain logic. Examples are the systems **D**, **T**, **S4** or **S5**.

2.2.2 Deontic logic

Temporal logic has operators that concern *time* and doxastic logic has operators that concern *believe*. When looking at legal problems and law, operators that concern norms are most suited [14]. We can find these operators in deontic logic, the logic concerning *ethics*, which consists of these three operators [30]:

1. **O** for "It is obligatory that ..."
2. **P** for "It is permissible that ..."
3. **F** for "It is forbidden that ..."

Deontic- and alethic logic have essential similarities, this can for example be seen in the fact that they have a basis is the same principles. For example, deontic logic also has five normative statuses [19]:

1. It is obligatory that ...
2. It is permissible that ...
3. It is impermissible that ...
4. It is omissible that ...
5. It is optional that ...

The first three actually resemble the three operators given before. The fourth however is often ignored and the fifth status is often described as: "It is a matter of indifference that ...", and thus defined in terms of the first three statuses. Like in alethic logic, the statuses can also be defined in terms of the first status, again r is used as a proposition:

1. $\mathbf{O(r)}$: $\mathbf{O}r$
2. $\mathbf{P(r)}$: $\neg\mathbf{O}\neg r$
3. $\mathbf{F(r)}$: $\mathbf{O}\neg r$
4. $\neg\mathbf{O}r$
5. $\neg\mathbf{O}r$ & $\neg\mathbf{O}\neg r$

Figure 2.2 shows the Deontic Square (DS), which is similar to the modal square of opposition. The status for optionality can also be added to this square to obtain the deontic hexagon in figure 2.3.

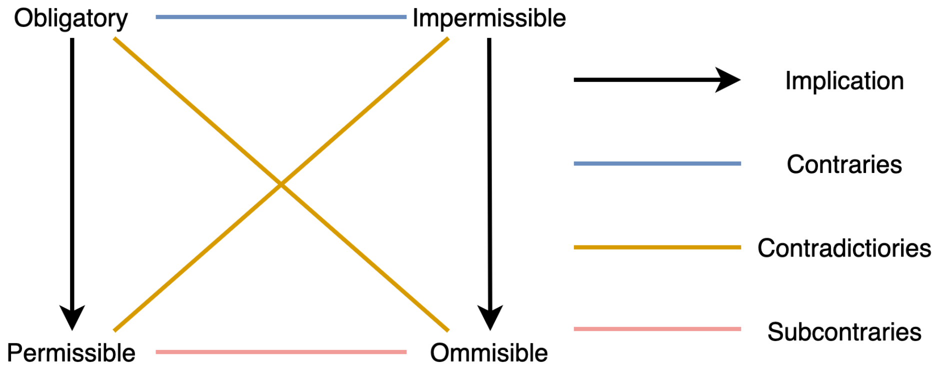

Figure 2.2: Deontic Square (DS) [19])

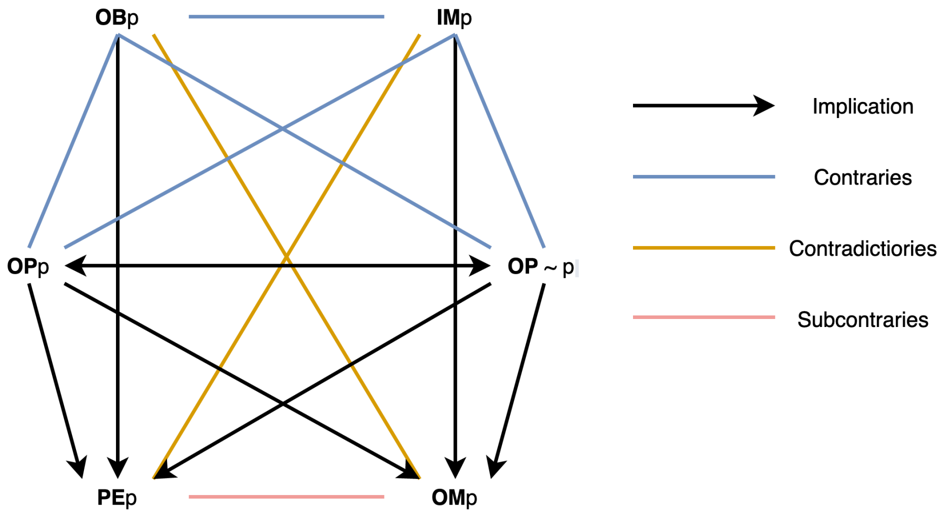

Figure 2.3: Deontic hexagon [19])

In figure 2.3 abbreviations of the statuses are used instead of the full words. p is used as a proposition:

- **OB**p: It is obligatory that p.
- **IM**p: It is impermissible that p.
- **OP** $\sim p$: It is not optional that p.
- **OM**p: It is omissible that p.
- **PE**p: It is permitted that p.
- **OP**p: It is optional that p.

As said before, often systems are added to the logic **K** to make it more suitable for a certain logic. To construct deontic logic an axiom **D** is often added to the logic **K**, this is then called the logic **D**. The axiom **D**, where A is a formula:

$$\mathbf{O}A \to \mathbf{P}A$$

This axiom states that what is obligatory is also permitted. The other way around however is not true, since all that is permitted is not necessarily obligatory. This system can also be used in other modal logics, for example in alethic modal logic. But there are differences between alethic modal logic and deontic logic. The system **T** (some logicians call this system **M**) adds the following axiom:

$$\Box A \to A$$

In alethic modal logic this axiom states that anything that is necessary is true. In deontic logic this axiom states that anything that is obligatory is true, but this is not the case. There are things that are obligatory but not done.

2.2.3 Standard Deontic Logic & Propositional Deontic Logic

The basis of this subsection mostly comes from the doctoral thesis by Royakkers [22]. Since it is a fairly large thesis only the parts needed to solve the two cases used in this thesis are elaborated on below. Reading the doctoral thesis and the adaptation by Royakkers [24, 23] fully will cover more topics that are beyond the scope of this thesis. Because the doctoral thesis by Royakkers [22] is used extensively in this thesis some definitions and explanations are the same in this thesis. This is because changing those definitions and explanations would alter the meaning, and thus making them incorrect.

In Standard Deontic Logic (SDL) there cannot be dealt with actions, a norm is expressed by applying a operator (**O**) to letters (for example: p) for example $\mathbf{O}(p)$ means: "It is obligatory that p". This cannot be read as: "It is obligatory to do p", this shows the difference between Ought-to-do and Ought-to-be statements. Ought-to-do statements are those that involve both agents and actions and support imperatives. These can change the state of a current situation or world. Ought-to-be statements are those that involve state of affairs and assertions, thus not agents and actions, and thus cannot change the current situation. SDL cannot deal with Ought-to-do statements, and thus cannot represent all legal rules in the correct manner. Propositional Deontic Logic PD_eL, however, can deal with Ought-to-do statements. Therefore SDL will be used for Ought-to-be statements and PD_eL will be used for Ought-to-do statements. Below SLD and PD_eL are elaborated on.

Standard Deontic Logic

The most studied system of deontic logic is Standard Deontic Logic (SDL) [19]. SDL builds upon propositional logic, and it uses the three deontic operators given before ($\mathbf{O}, \mathbf{P}$ and $\mathbf{F}$) [22]. Normative judgements can be formed from combining propositions and the operators, these are called well-formed formulas (wff's) of the system. SDL is the logic $\mathbf{D}$ added with suggestive notation to express the intended interpretation [19]. SDL is axiomatised by the rule of inference, p and q are propositions [22]:

$$\textbf{OB-RE}: \quad \frac{p \leftrightarrow q}{O(p) \leftrightarrow O(q)}$$

RE is the rule of inference for equivalence, it states that equivalent propositions are equally necessary:

$$\textbf{RE}: \quad \frac{p \leftrightarrow q}{\Box p \leftrightarrow \Box q}$$

OB-RE is the rule of inference for equivalence (RE) applied in the logic $\mathbf{D}$ with the operator for obligation ($O()$).OB-RE states that if $p \leftrightarrow q$ is a formula, then so is $O(p) \leftrightarrow O(q)$. This roughly states that if two formulas are provably equivalent then the results of prefixing them with O is too [19]. SDL has the following axiom schema:

Axiom 2.2.1.

1. **OB-M**: $(\mathbf{O}(p) \wedge \mathbf{O}(q)) \rightarrow \mathbf{O}(p \wedge q)$
2. **ON**: $\mathbf{O}(p \vee \neg p)$
3. **OD**: $\neg\mathbf{O}(p \wedge \neg p)$
4. **Df.P**: $P(p) \equiv \neg\mathbf{O}(\neg p)$

OB-M is similar to the distribution axiom, it states that if it is obligatory that p and it is obligatory that q then it is obligatory that p and q. ON states that it is obligatory that p or not p, which is intuitive, since it is obligatory for something to be true or not true. Following ON, OD states that it is not obligatory that p and not p. ON and OD together make it that something is either obligatory or not, not both. Df.P is a definition that shows the similarity of "it is permitted that p" and "it is not obligatory that not p".

With SDL, standard Kripke-style possible world semantics are used. The following model structure is used: $\mathcal{M} = (W, \mathcal{R}, V)$.

1. $W = \{w_1, w_2, ...\}$
   The set of possible worlds.
2. $R(w) \rightarrow \mathcal{P}(W)$
   The accessibility function $R \in \mathcal{R}$, it takes a world w and returns a subset of W.

3. $V(w,p) \to \{\text{true}, \text{false}\}$ with p being an atomic proposition and $w \in W$. A valuation function which assigns true or false to a proposition p at a world in W.

The accessibility function R yields the deontically ideal worlds relative to a given world. A deontically ideal world is a world where every obligation is never neglected and everything permissible is the case sometimes [29]. Following these axioms and the semantics, truth conditions for both O and P can be defined. The function: $[\![]\!] \in L \to 2^W$ is used. L is the set of all wff's in propositional logic.

$$(1) \quad \mathcal{M}, w \vdash O(p) \text{ iff } R(w) \subseteq [\![p]\!]$$

$O(p)$ holds in w iff p is true in all ideal worlds with respect to w.

$$(2) \quad \mathcal{M}, w \vdash P(p) \text{ iff } R(w) \cap [\![p]\!] \neq \emptyset$$

$P(p)$ holds in w iff p is true in at least one ideal world with respect to w.

SDL with actors: In SDL obligation and permission is currently only focused on one actor and its obligations and permissions. We cannot say that acts are obligatory for one particular individual or a particular group of individuals and not for others. It also cannot be said that an act is obligatory for some, but not all members of a group of individuals. The first step in adding actors to SDL is being able to say: "It is obligatory for i that p". This is done by the revitalised obligation: $O_i(p)$, this is a personal obligation. The personal permission being $P_i(p)$. There are three sorts of obligations that bind an individual:

1. **Personal obligation**: obligation for a specific individual.
2. **General obligation**: obligation for every individual.
3. **Unspecific obligation**: obligation for some individuals.

The following model structure is used: $\mathcal{M} = (W, I, \mathcal{R}, V)$. It is an extension of the model structure given for SDL before.

1. $W = \{w_1, w_2, \dots\}$
   The set of possible worlds.
2. $I = \{i_1, i_2, \dots\}$
   The set of individuals.
3. $\mathcal{R} = \{R_i | i \in I\}$
   Given a world $R_i : W \to 2_W$, the function $R_i \in \mathcal{R}$ on W returns the deontically ideal worlds for individual i.

4. $V(w, p) \rightarrow \{\text{true}, \text{false}\}$ with $w \in W$ and p being an atomic proposition. A valuation function which assigns true or false to a proposition at a world in W.

The truth conditions can again be defined:

$$\text{(1)} \quad \mathcal{M}, w \vdash O_i(p) \text{ iff } R_i(w) \subseteq [\![p]\!]$$

$$\text{(2)} \quad \mathcal{M}, w \vdash P_i(p) \text{ iff } R_i(w) \cap [\![p]\!] \neq \emptyset$$

The following restraint also needs to be added:

$$R_i(w) \neq \emptyset, \text{ for all } R_i \in \mathcal{R} \text{ and for all } w \in W.$$

Now we can define general obligation, general permission, unspecific obligation and unspecific permission as follows:

- The general obligation: $\forall_{i \in I} O_i(p) \equiv O^+(p)$;
- The general permission: $\forall_{i \in I} P_i(p) \equiv P^-(p)$;
- The unspecific obligation: $\exists_{i \in I} O_i(p) \equiv O^-(p)$;
- The unspecific permission: $\exists_{i \in I} P_i(p) \equiv P^+(p)$.

The following principles are valid:

- $O^+(p) \rightarrow O^-(p)$
  General obligation implies unspecific obligation;
- $P^-(p) \rightarrow P^+(p)$
  Unspecific permission implies general permission;
- $O^+(p) \rightarrow P^-(p)$
  General obligation implies unspecific permission;
- $O^-(p) \rightarrow P^+(p)$
  Unspecific obligation implies general permission;
- $O^-(p) \rightarrow P_i(p)$
  Unspecific obligation implies personal permission;
- $O_i(p) \rightarrow P^-(p)$
  Personal obligation implies unspecific permission.

The last addition we make to SDL in this thesis is directed obligation. The model structure that is used is: $\mathcal{M} = (w, \mathcal{P}^+(I), \mathcal{R}'_I, V)$. It is an extension of the model structure given for SDL before.

1. $W = \{w_1, w_2, \dots\}$
   The set of possible worlds.

2. The powerset $\mathcal{P}^+(I)$ of the set $I = \{i_1, i_2, \dots\}$

3. $\mathcal{R}'_I = \{\mathcal{R}_{X,Y} | X, Y \in \mathcal{P}^+(I)\}$.
   The function $\mathcal{R}_{X,Y} \in \mathcal{R}'_I$ on W returns the deontically ideal worlds for X towards Y given a world: $\mathcal{R}_{X,Y} : W \to 2^W$.

4. $V(w, p) \to \{\text{true}, \text{false}\}$
   A valuation function which assigns true or false to a proposition p to $w \in W$.

The truth condition can be defined as follows:

$$(1) \quad \mathcal{M}, w \vdash {}_Y O_X(p) \text{ iff } \mathcal{R}_{X,Y}(w) \subseteq [\![p]\!]$$

The following restraint again needs to be added (after adjustment):

$$\mathcal{R}_{X,Y}(w) \neq \emptyset, \text{ for all } \mathcal{R}_{X,Y} \in \mathcal{R}'_I \text{ and for all } w \in W.$$

Directed obligation is the obligation someone has to someone else, also called duty. "it is obligatory for X towards Y that p" is denoted as: ${}_Y O_X(p)$ with $X, Y \in P^+(I)$.

Propositional Deontic Logic

The basis of PD_eL is the logic framework of (propositional) dynamic logic. Dynamic logic is logic extended to reason about more complex behaviour, for example the behaviour of agents and actions which we need for Ought-to-do statements.

To reduce deontic operators to dynamic operators, a violation atom V is used, it indicates that an action took place that violated one of the deontic constraints. This means that the performance of a forbidden action leads to a bad state of affairs, e.g. a sanction. Several interpretations of V are possible, the interpretation that is used in Royakkers his doctoral thesis [22] and which will be used in this thesis is: V equals the situation that is in contravention of the law, or stated in other words: V equals the situation that breaches the law. Whether it leads to a sanction and which sanction is given is left aside. In PD_eL propositional language is extended with a modal operator: $[\beta]$ for every action β in the language. $[\beta]\Phi$ means that after β is performed, assertion Φ holds. Now since deontic logic is described as a variant of dynamic logic, actions and assertions can be described strictly separated.

A is the set of action symbols. $\underline{a}$ is an atomic action, an atomic action is an action which cannot be interrupted while the action takes place [20]. From this it follows that $\underline{A}$ is the set of atomic actions. To form the set of semantic elementary actions two symbols are added that are not in A: *skip* and δ. The *skip* action expression is the action that has no effect, or the empty action. The δ action expression is also denoted as *fail*, this is the action expression that expresses that that action always fails. After this action the system stops. Furthermore in A are the following action expressions: $\beta_1 \cup \beta_2$, $\beta_1 \wedge \beta_2$, $\beta_1;\beta_2$, $\bar{\beta}$, *any* and *change*.

- $\beta_1 \cup \beta_2$ means the choice between β_1 and β_2.
- $\beta_1 \wedge \beta_2$ means the simultaneous performance of β_1 and β_2.
- $\beta_1;\beta_2$ means the sequential composition of β_1 and β_2.
- $\bar{\beta}$ means the negation of action expression β
- *any* indicates that it does not matter which action is represented by the action expression.
- *change* indicates that is does not matter which action is represented by the action expression, as long as it is not the *skip* action.

This defines the following Backus-Naur form(BNF):

$$\beta ::= \underline{a}|\beta_1 \cup \beta_2|\beta_1 \wedge \beta_2|\beta_1;\beta_2|\bar{\beta}|\ \mathit{any}\ |\ \mathit{fail}\ |\ \mathit{skip}\ |\ \mathit{change}$$

Now the language *Act* of PD_eL can be discussed. *Act* consists of assertions concerning action expressions, defined by the following BNF:

$$\Phi ::= \phi|\Phi_1 \wedge \Phi_2|\Phi_1 \vee \Phi_2|\Phi_1 \rightarrow \Phi_2|\neg\Phi|[\beta]\Phi$$

ϕ is a propositional variable in L, which is the language of propositional logic. Now the deontic operators can also be expressed using these notions:

- $F(\beta) \equiv [\beta]V$
- $P(\beta) \equiv \neg[\beta]V$
- $O(\beta) \equiv [\bar{\beta}]V$

$F(\beta)$ meaning it is forbidden to perform β, thus if β is performed you are in violation. $P(\beta)$ meaning that it is permissible to do β, thus you are not in violation when β is performed. $O(\beta)$ meaning that it is obliged that β is performed, thus you are in violation when β is not performed.

Semantics for action expressions Synchronicity sets (s-sets) are sets of sequences which give the semantics for action expressions.

Definition 2.2.1. A s-set denotes a set of elementary actions that are performed simultaneously. Every non-empty subset of A is a synchronicity set, as well as $[\delta]$ and $[skip]$. $S, S_1, S_2, ...$ denote s-sets. $\mathcal{S}$ denotes the powerset of s-sets with actions in A.

$[\delta]$ and $[skip]$ are not in an $\mathcal{S}$ since they cannot be performed simultaneously with actions in A since it is not possible to do nothing and perform an action at the same time.

The course of performances of actions is denoted by a synchronicity sequence (s-sequence), which is a sequence of s-sets. The definition of a synchronicity sequence is:

Definition 2.2.2. A synchronicity sequence is a finite or infinite sequence of $S_1 S_2 \dots S_n \dots$ of s-sets. Within a synchronicity sequence, ϵ stands for the empty sequence. $[\delta]$ can only be the last s-set of an s-sequence. $l(t)$ is the number of s-sets in a s-sequence t.

Two s-sequences can be concatenated by the $\circ$ operator.

Definition 2.2.3. Let $t = S_1 \dots S_n$ and $t' = S'_1 \dots S'_m$ be two sequences, then,

$$t \circ t' = \begin{cases} t, & \text{if } S_n = [\delta] \\ S_1 \dots S_n S'_1 \dots S'_m & \text{if } S_n \neq [\delta] \end{cases}$$

If t is an infinite s-sequence, then $t \circ t' = t$.
$t \circ \epsilon = t$
$[\delta] \circ t = [\delta]$

$T, T_1, T_2, \dots$ are used to denote sets of s-sequences. Since the language of action expressions is non-deterministic, the sets of s-sequences have to be considered as the semantics of an action expression. $\underline{\text{a}}$, an atomic action, indicates that there is a choice between all possible actions in which a is at least performed. In such a set, each s-sequence is a possible choice. The domain A is defined as follows:

Definition 2.2.4. $\mathcal{A}$ is the collection of sets T consisting of s-sequences.

To give the notation for all action expressions in Act, we need to define the following operators: $\sqcup, \sqcap$ and $\sim$. $\sqcup$ resembles the $\cup$ of sets.

Definition 2.2.5. For $T, T' \in \mathcal{A}$:

$$T \sqcap T' = \begin{cases} T \cap T', & \text{if } T \cap T' \neq \emptyset \\ \{[\delta]\}, & \textit{otherwise} \end{cases}$$

$\sqcap$ resembles the $\cap$ of sets. Needed to define $\sqcup$ is the following definition:

Definition 2.2.6. Let T be a set of s-sequences, then:

$$T^\delta = \begin{cases} T \setminus \{[\delta]\}, & \text{if } \exists_{S \in T} S \neq [\delta] \\ \{[\delta]\}, & \textit{otherwise} \end{cases}$$

The idea of this operator is that when there is a non-failing alternative, failure is avoided. Now using definition 2.2.6, $\sqcup$ can be defined:

Definition 2.2.7. For $T, T' \in \mathcal{A}$:

$$T \sqcup T' = (T \cup T')^{\delta}$$

The operator $\sim$ has two clauses, the first clause expresses that an s-set $S \neq [\delta]$ is not involved, by considering all other s-sets in $\mathcal{S}$, and any s-set is possible when one does not fail. The second clause expresses that $\widetilde{T}$ is the negation of a set T by taking the $\sqcap$-intersection of the s-sets S in T. The operator $\sim$ is defined as follows:

Definition 2.2.8.

1. For an s-set S,

$$\widetilde{S} = (\mathcal{S} \cup \{[skip]\}) \setminus S$$

2. For a non-empty set $T \in \mathcal{A}$,

$$\widetilde{T} = \sqcap\{\widetilde{S} | S \in T\}$$

Now the semantic function $[\![]\!] \in Act \to \mathcal{A}$ can be defined:

Definition 2.2.9.

1. $[\![a]\!] = \{S \in \mathcal{S} | a \in S\}$
2. $[\![\beta_1 \cup \beta_2]\!] = [\![\beta_1]\!] \sqcup [\![\beta_2]\!]$
3. $[\![\beta_1 \cap \beta_2]\!] = [\![\beta_1]\!] \sqcap [\![\beta_2]\!]$
4. $[\![\beta]\!] = [\![\widetilde{\beta}]\!]$
5. $[\![skip]\!] = \{[skip]\}$
6. $[\![fail]\!] = \{[\delta]\}$
7. $[\![any]\!] = \mathcal{S} \cup \{[skip]\}$
8. $[\![change]\!] = \mathcal{S}$

The PD_eL used in this thesis (as well as in the doctoral thesis by Royakkers) is axiomatised by the following rules of inference:

$$(N) \quad \text{if} \vdash \Phi, \text{then} [\beta]\Phi$$

(N) means that if an assertion can be derived from the system, then the assertion cannot be revoked by any action.

$$(S) \quad \beta_1 =_A \beta_2 \vdash [\beta_1]\Phi \equiv [\beta_2]\Phi$$

(S) is the notion that the equality of action expressions can be denoted in terms of their semantics.

In PD_eL the following axioms hold:

Axiom 2.2.2.

1. *All tautologies of propositional logic*
2. $[\beta](\Phi_1 \rightarrow \Phi_2) \rightarrow ([\beta]\,\Phi_1 \rightarrow [\beta]\,\Phi_2)$
3. $[\beta_1 \cup \beta_2]\,\Phi \equiv [\beta_1]\,\Phi \wedge [\beta_2]\,\Phi$
4. $[\beta_1]\,\Phi \vee [\beta_2]\,\Phi \rightarrow [\beta_1 \wedge \beta_2]\,\Phi$
5. $[fail]\,\Phi$
6. $[skip]\,\Phi \equiv \Phi$

Since PD_eL has its basis in propositional logic, the tautologies of propositional logic are also an axiom in PD_eL.

Axiom 2.2.2.2 states that after the performance of action β, if assertion Φ_1 holds then assertion Φ_2 holds, it is equivalent to: if Φ_1 holds after action β is performed then Φ_2 holds after action β is performed.

Axiom 2.2.2.3 states that assertion Φ holds after the action you choose to perform, either action β_1 or β_2, is performed. It is equivalent to performing action β_1 after which the assertion Φ holds and performing action β_2 after which the assertion Φ holds. The performance of either of the actions leads to the same assertion.

Axiom 2.2.2.4 states that Φ holds after performing one of the actions β_1 or β_2, it is equivalent to Φ holding after performing both actions β_1 and β_2. Performing both actions does not change assertion Φ.

Axiom 2.2.2.5 states that when one has to perform the impossible action, there are no successor worlds and thus assertion Φ holds in all worlds.

Axiom 2.2.2.6 states that performing the action *skip* after which assertion Φ holds is equivalent to Φ because the action *skip* has no effect on Φ.

The propositions of PD_eL can be found in the appendix section: A.1. A model M for PD_eL is given by: $M = (A, W, [\![\beta]\!]_R, \pi)$:

1. $A = \{\beta_1, \beta_2, \dots\}$
   The set of actions.
2. $W = \{w_1, w_2, \dots\}$
   The set of possible worlds.
3. $[\![\beta]\!]_R$ is a function that associates with action $\beta \in Act$ and world w, it returns the set of possible worlds which the performance of β leads to.
4. $\pi : W \times L \rightarrow \{\text{true}, \text{false}\}$
   The truth relation between world and sentences.

PD_eL **with actors** : Unlike SDL, in PD_eL the distinction is not whether it is a personal obligation or general obligation, but more whether it is an action performed by one actor or an action performed by a group of actors. In the cases we elaborate on in this thesis, only one actor performs the action thus that is the part that is elaborated on.

For an actor $i \in I$ and an action $\beta \in Act$, the atomic action is defined as follows: $i : \beta$ which states: i performed action β. $\underline{Evt}$ is the set of atomic events, Evt is the set of all event expressions which is defined by the following BNF, with $\alpha \in Evt$:

$$\alpha ::= i : \beta | \alpha_1 \cup^* \alpha_2 | \alpha_1 \wedge^* \alpha_2 | \overline{\alpha}$$

- $\alpha_1 \cup^* \alpha_2$ means the choice between α_1 and α_2.
- $\alpha_1 \wedge^* \alpha_2$ means the simultaneous performance of α_1 and α_2.
- $\bar{\alpha}$ means the negation of action expression α
- $\{i : any\}$ indicates that i performs a universal action.
- $\{i : change\}$ indicates that i performs a universal action, that is not the *skip* action.
- $\{i : skip\}$ indicates that i performs an empty action, thus nothing happens.
- $\{i : fail\}$ indicates that i performs an action that fails, after this action the system also stops.

The semantics of event expressions are similar to the semantics defined before, they are again expressed by synchronicity sets (s-sets). Here the s-sets denote performances of packages of elementary actions that have to be performed simultaneously by the same actor.

Definition 2.2.10. $\{\delta\}, \{i : skip\}$ and every pair of a non-empty subset of A, with A being the set of actions, and an individual i in I, with I being the set of individuals, are s-sets.

s-sets are denoted by $S, S_1, S_2, \ldots$. The set of all s-sets, except $\{\delta\}$, is denoted by $\mathcal{S}^*$.

Definition 2.2.11. Let $I = \{i_1, i_2, \ldots, i_n\}$, then $\mathcal{T}$ is defined as the set of elements of the indirect product of $\mathcal{S}^*_{i,1}, \mathcal{S}^*_{i,2}, \ldots, \mathcal{S}^*_{i,n}$, this is denoted as: $\mathcal{T} =_{def} \times_{i \in I} \mathcal{S}^*_i$. With the indirect product an $i \in I$ is matched to only one $\mathcal{S}^*$, instead of each $i \in I$ begin matched to every $\mathcal{S}^*_{i,1}, \mathcal{S}^*_{i,2}, \ldots, \mathcal{S}^*_{i,n}$ (the direct product). An element of $\mathcal{T}$ is called a step and denoted as: t. t_i is the set of actions in A, in the s-set t of actor i.

t denotes the deterministic set of actions for all actors simultaneously. Because of the *skip* operator it is possible for an individual to perform an action with no outcome, this is needed because not all the time all actors affect the situation. $T, T_1, T_2, \ldots$ are used to denote sets of steps. The domain $M_{\mathcal{T}}$ is defined as follows:

Definition 2.2.12. $M_{\mathcal{T}}$ is the collection of sets T consisting of steps. $M_{\mathcal{T}}$ is the powerset of $\mathcal{T}$.

Here the operators $\sqcup^*, \sqcap^*$ and $\sim$ are used as semantic counterparts for the $\cup^*, \wedge^*$ and $-$ syntactical operators defined in the BNF of *Evt*. $\sqcup^*$ is the counterpart for $\cup^*$, $\sqcap^*$ is the counterpart for $\wedge^*$ and $\sim$ is the counterpart for $-$.

Definition 2.2.13. For $T_1, T_2 \in M_{\mathcal{T}}$:

$$T_1 \sqcap^* T_2 = \begin{cases} T_1 \cap T_2, & \text{if } T_1 \cap T_2 \neq \emptyset \\ \{[\delta]\}, & \textit{otherwise} \end{cases}$$

Using definition 2.2.6, $\sqcup^*$ can be defined:

Definition 2.2.14. For $T_1, T_2 \in M_{\mathcal{T}}$:

$$T_1 \sqcup^* T_2 = (T_1 \cup T_2)^{\delta}$$

The operator $\sim^*$ has like $\sim$ two clauses, the first clause expresses the negation of t, by taking the set-theoretic complement of $\{t\}$ with respect to T. The set-theoretic complement are the elements of T that are not in $\{t\}$ The second clause expresses the negation of T, this is done by taking the negation of each t in T, and then taking the intersection $\sqcup^*$ of that. The operator $\sim$ is defined as follows:

Definition 2.2.15.

1. For an step t,

$$t^{\sim^*} = \mathcal{T} \setminus t$$

2. For a non-empty set $T \in M_{\mathcal{T}}$,

$$T^{\sim^*} = \sqcap^* \{t^{\sim^*} | t \in T\}$$

Now the semantic function $[\![]\!] \in Evt \to M_{\mathcal{T}}$, with $\underline{a}, \beta_l, \beta_2 \in Act$, $i \in I$ and $\alpha, \alpha_l, \alpha_2 \in Evt$, is given by:

1. $[\![i : a]\!] = \{t \in \mathcal{T} | a \in t_i\}$;
2. $[\![i : \beta_1 \cup \beta_2]\!] = [\![i : \beta_1]\!] \sqcup^* [\![i : \beta_2]\!]$;

3. $[\![i : \beta_1 \wedge \beta_2]\!] = [\![i : \beta_1]\!] \sqcap^* [\![i : \beta_2]\!]$;

4. $[\![i : \overline{\beta_1}]\!] = [\![i : \widetilde{\beta_1}]\!]$

5. $[\![\alpha_1 \cup^* \alpha_2]\!] = [\![\alpha_1]\!] \sqcup^* [\![\alpha_2]\!]$;

6. $[\![\alpha_1 \wedge^* \alpha_2]\!] = [\![\alpha_1]\!] \sqcap^* [\![\alpha_2]\!]$;

7. $[\![\overline{\alpha_1}]\!] = [\![\alpha_1]\!]^{\sim^*}$

8. $[\![i : fail]\!] = \{[\delta]\}$;

9. $[\![i : any]\!] = \mathcal{T}$

10. $[\![i : skip]\!] = \{t \in \mathcal{T} | t_i = skip\}$;

11. $[\![i : change]\!] = \mathcal{T} \setminus [\![i : skip]\!]$.

A model M for $PD_eL(Evt)$ is given by: $M = (I, A, W, [\![\alpha]\!]_R, \pi)$:

1. $I = \{i_1, i_2, \ldots, i_n\}$
   The set of actors.

2. $A = \{\beta_1, \beta_2, \ldots\}$
   The set of actions.

3. $W = \{w_1, w_2 \ldots\}$
   The set of possible worlds.

4. $[\![\alpha]\!]_R$ is a function that associates with event $\alpha \in Evt$ and world w, it returns the set of possible worlds which the performance of α leads to.

5. $\pi : WxL \rightarrow \{\text{true}, \text{false}\}$
   The truth relation between world and sentences.

Lastly, here too we add directed obligation: $O(i : \underline{a(j)})$ which can be read as: "i has a duty towards j", the duty is a().

2.3 Reduction Graph representation

A reduction graph is a representation of subgoals that arise in the process of constructing an inference path from the predicate to be established to the facts that belong to the case [7].

A reduction graph is build like an AND/OR graph [6]. Both AND/OR trees and AND/OR graphs exist. To represent the full variety of possible situations that can occur in problem reduction an AND/OR graph is used. In an AND/OR tree each node has at the most one parent, in an AND/OR graph this is not the case. In an AND/OR graph reasoning proceeds backwards from the initial goal, which is the initial problem that is to be solved.

Each node in an AND/OR graph represents a problem or a set of problems that needs to be solved. At first an initial problem description is given, then this problem is solved by a sequence of transformations that, in the end, change into a set of subproblems whose solutions are immediate. All subproblems need to be solved to solve the initial problem. There are several rules as to which the AND/OR graph is constructed:

1. Each node in the graph either represents a single problem or represents a set of problems to be solved.

2. The graph contains a start node which is the original problem.

3. A terminal node is a node representing a primitive problem. A primitive problem is a problem that can be solved immediate. A terminal node has no descendants.

4. To transform a problem P into a set of subproblems an operator is applied to P. This is shown in the graph as a directed arc from P to a node representing the set of subproblems.

5. When a set of (sub)problems A can only be solved when all of its members (B, C) can be solved it is called an AND-node. This is shown by a horizontal line crossing through the arcs leading from an AND-node to its successors. Figure 2.4 shows an example where A is a set of subproblems and, B and C are the subproblems.

6. When a problem P can be solved by any of its successors, P is called an OR-node. Figure 2.5 shows an example where P is a set of subproblems and, Q and R are subproblems.

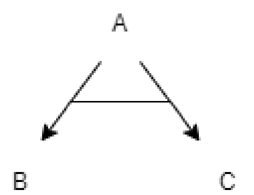

Figure 2.4: Example of an AND-node

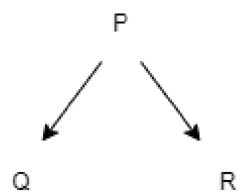

Figure 2.5: Example of an OR-node

Whether a node is an AND-node or an OR-node can be seen in the formula, when it is a formula with a conjunction in it, it will be an AND-node. If the formula has a disjunction, it will be an OR-node.

With only a constructed AND/OR graph the problem is not yet solved. To find a solution for the initial problem an AND/OR graph with enough nodes such that the start node can be solved. An AND/OR graph containing only the nodes needed to solve the start node is called a solution graph. There

are also several rules that determine if a node is solvable or not.
A node is solvable if:

- A node is a terminal node;
- A node is a non-terminal node that has AND-nodes as successors and those successors are all solvable;
- A node is a non-terminal node that has OR-nodes as successors and at least one of those successors is solvable.

From these rules the rules for a node that is not solvable can be derived. A node is not solvable if:

- A node is non-terminal but also has no successors.
- A node is non-terminal and had AND-nodes as successors but at least one of the successors is not solvable.
- A node is non-terminal and has OR-nodes as successors but all OR-nodes are not solvable.

To use AND/OR graphs, or as they are sometimes called: reduction graphs, for representing legal cases several aspects of the graph need to be different. In a reduction graph representing a case, the reduction graph represents a justification of why a certain predicate is true or not. To construct a reduction graph representing a case, legal rules (the law), legal precedents and the facts of the case are used. A legal precedent is a past case in which a court resolved a legal problem, there are also several general precedents like: reasonable care, malice, activity in furtherance of employment. In the law an important principle is *Stare Decisis*, this means that the court is obliged to follow historical cases when making a ruling on a similar case [32]. A reduction graph representing a legal case takes both the legal rules (laws) and the legal precedents into account. In the law arguments can differ in strength, thus a ranking is often needed. This means that an argument with a higher strength will be higher in the ranking and thus be considered with more weight to it. To construct a reduction graph concerning a legal case, the following input is required:

1. A set of facts.
2. A collection of legal authorities:
   - Warrants which occur as reduction-operators.
   - Legal rules coming from sources other than from precedents.
   - Additional knowledge.
3. A goal proposition which is the initial problem.

4. An evaluation criteria such that the arguments can be ranked by strength.

When this input is collected, two things need to be done:

1. Find each of the reduction graph presentations possible for goal proposition and its negation.

2. Rank the reduction graphs on the basis of the evaluation criteria established before.

Using an example from the paper by Branting [7] as a guideline, the manner in which a reduction graph concerning a legal case is built is elaborated on. The example is the fictional Adam v Baker case. A small summary of the case:

Adam and Baker played a hockey game together. Adam intentionally uses his hockey stick to hit Baker's hockey stick to prevent Baker from hitting the puck. Baker ended up spraining his thumb causing 1.000 dollars in medical bills. Baker sued Adam for battery, which is unauthorised application of force against another person's body, which results in offensive touching or actual physical injury [16].

The requirements to establish a claim for battery are (1) harmful or offensive touching and (2) no consent. Which can also be shown as:

$$\text{battery} \leftrightarrow \text{touching} \wedge\neg \text{ consent}$$

Since a hockey player consents to contacts that are custom during a hockey match, Adam is not liable to Baker for battery. Thus the judgement against Adam is nullified. In figures 2.6 and 2.7 the reduction graph representations of the case are shown.

First figure 2.6, the reduction graph claiming that battery is the case. In this reduction graph *battery* is the original problem, the starting node. As given before, for battery to be true both touching and not consent need to be "solved". Therefore *battery* is an AND-node with the nodes *touching* and $\neg$ *consent* as successors. Going down from *touching*: for *touching* to be true, *hitting hockey stick* needs to be solved. This matches the facts in the case, and thus ends in a terminal node with no descendants. However, when the $\neg$ *consent* node needs to be solved, this ends up being a non-terminal node with no descendants. Thus we cannot solve this reduction graph.

Second, figure 2.7. In this reduction graph it can be seen that $\neg$*battery* is the original problem, the starting node. As given before, for $\neg$*battery* to be true either, both *touching* and *consent* need to be "solved" or $\neg$*touching* and $\neg$*consent* need to be "solved". It can be seen in the previous reduction graph (figure 2.6) that *touching* can be solved. We thus will not try to solve battery with a node $\neg$touching. The nodes *touching* and *consent* are used, since they both need to be true, $\neg$*battery* is an AND-node with *touching* and *consent* being the successors. Going down from *touching*, for touching to be true, *hitting hockey stick* needs to be solved. This matches the facts in the case, and thus ends in a terminal node with no descendants. This arc is therefore solved. Going down from *consent*, for consent to be true, sub-problem set *participating in hockey game* needs to be solved. This arc can then also be solved by the facts of the case, which again is a terminal node with no descendants. Since both sides are true and $\neg$*battery* is an AND-node, $\neg$*battery* is true.

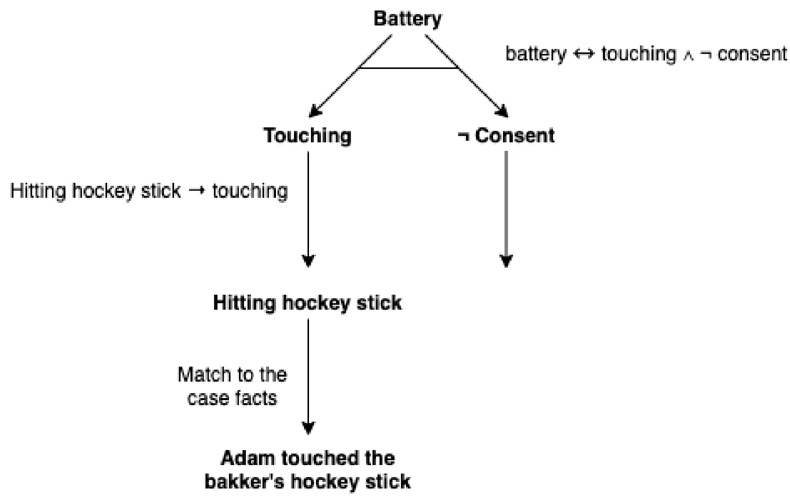

Figure 2.6: Reduction-graph representation of Adams v Bakker claiming that battery is the case

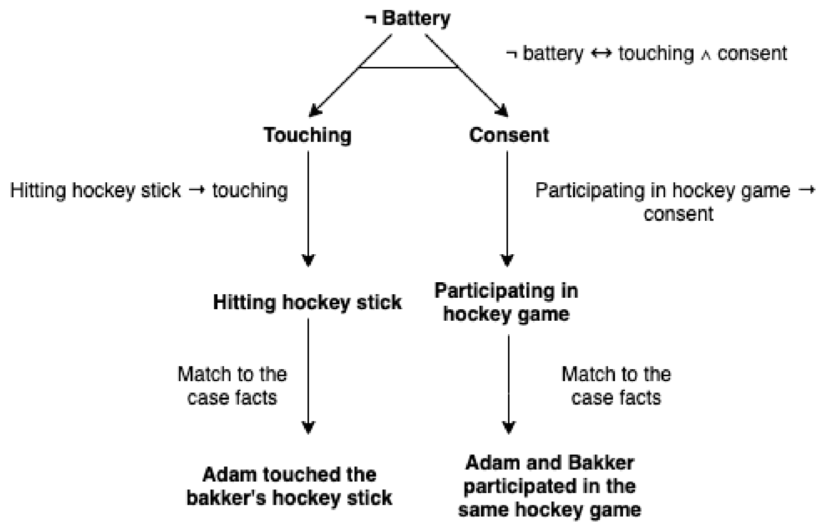

Figure 2.7: Reduction-graph representation of Adams v Bakker claiming that battery is not the case

Now also from the reduction graph representation it can be seen that battery cannot be claimed. Thus the outcome of the model matches the outcome of the Adam v Bakker case.

Chapter 3

The Logical Models

For this thesis two cases will be used to be able to compare both forms of logical models, one case is taken from the paper by Branting [7]. This is the Bourhill v Young case. The other case is taken from the doctoral thesis by Royakkers [22]. This case has no specific name, but in this thesis this will be called: HI's case. These two cases are used because in the paper by Branting, the Bourhill v Young case is already worked out using the reduction graph representation. In the doctoral thesis by Royakkers, HI's case is already worked out using $PDeL$. Therefore we may assume that these are correct. In this thesis the cases are explained and the models already worked out are worked out again using the papers as guidance or example. For the cases that are not worked out yet, the cases will be worked out in this thesis.

3.1 The cases used

3.1.1 HI's case

HI's case is a case that dates back to 1993. It is a case in which it is the question which rules have the upper hand. Royakkers used this case on several occasions [24, 22], there is however a citation mistake in both articles. To be able to find this case the following ECLI number needs to be used: ECLI:NL:HR:1993:ZC9571. Unfortunately, the High Court (Hoge Raad) in the Netherlands could not find this case in their archives, thus we have to use the facts stated in the articles by Royakkers [24, 22]. A small summary of the case written by Royakkers:

"On a national route road A28, within the city limits of Zwolle, a lorry from the firm H.I. drove at a speed of 96 km/h. H.I. was imposed an administrative sanction on the ground of a lorry exceeding the speed limit by 15 to 20 km/h. An appeal was lodged with the public prosecutor and the sub-district court judge, because H.I. was of the opinion that, on the road in question, traffic signs indicating a speed limit of 100 km/h were in force, and that, therefore, no sanctionable act had been committed, for traffic signs override traffic rules."[22].

In the Netherlands lorries have a maximum speed limit of 80 km/h. However, the Dutch traffic regulations state that traffic signs override traffic rules, in as far as these rules are incompatible with the signs. The judge ruled that according to the regulations there was no incompatibility. The sign stating that the speed limit is 100 km/h is a regulatory sign and not a sign that implies a higher speed for lorries. This can also be shown as a simplified formula:

ticket for the driver ↔ ¬traffic rules override signs ∧ the driver is a lorry driver ∧ lorries cannot drive faster than 80 km/h ∧ the driver drove 81 km/h or more

3.1.2 Bourhill v Young

Bourhill v Young is a relatively old case, it dates back to 1943. It is a well known case because it established important boundaries on the scope of recovery for bystanders. This is a summary of the case written by Branting:

"Young, a motorcyclist, was killed because of his own negligence when he passed a tram at excessive speed and collided with a car about 50 feet beyond the tram. At the time of the accident, the tram was stopped and Mrs. Bourhill was alighting. Mrs. Bourhill heard the collision and saw blood on the road after the accident and as a result suffered a nervous shock. Mrs. Bourhill was outside what Young ought to have contemplated as the area of potential danger that would arise from his careless driving, since she was alighting on the side of the tram opposite the side on which Young passed." [7].

In this case the question is whether the nervous shock from Mrs. Bourhill can be recovered on Young having a duty of care while driving to the people around him. Duty of care is a requirement that a person has towards others and the public to act with watchfulness, attention, caution and prudence that a reasonable person would use in the circumstances [15]. The case from Mrs. Bourhill against Young was dismissed. Young was not held liable because it was not foreseeable that Mrs. Bourhill would suffer harm because of a traffic accident caused by Young and because Mrs. Bourhill was about 50 feet away from the accident, thus not sufficiently proximate to the crash scene [18]. The requirements to establish a case negligence of duty of care are: (1) duty is present, (2) the duty must be breached and (3) proximate cause. This can be shown in the following formula:

negligence of duty of care ↔ duty ∧ breach of duty ∧ proximate cause

3.2 Deontic logic representation of the cases

3.2.1 HI's case

This case was originally worked out in the doctoral thesis by Royakkers [22]. The worked out case is used as a guideluine. For this case, some articles from the Reglement Verkeersregels en Verkeerstekens 1990 (RVV 1990) are needed. This is the Dutch regulation for traffic rules and traffic signs. The articles relevant to HI's case and thus needed are article 22 RVV 1990and article 63 RVV 1990:

- **Art. 22 RVV 1990** [1]:
  Voor zover niet ingevolge andere artikelen een lagere maximumsnelheid geldt, gelden voor de volgende voertuigen de volgende bijzondere maximumsnelheden:

  a. voor vrachtauto's, autobussen en motorvoertuigen met aanhangwagen 80 km per uur;
  b. voor tractoren en zelfrijdende werktuigen 25 km per uur.

  Which translates to:
  As far as in other articles a lower speed limit is not in order, the following maximum speed limits apply to vehicles:

  a. for lorries, busses and motor vehicles with a trailer 80 km per hour;
  b. for tractors and self-driving work vehicles 25 km per hour.

- **Art. 63 RVV 1990**[2]:
  Verkeerstekens gaan boven verkeersregels, voor zover deze regels onverenigbaar zijn met deze tekens.
  Which translates to:
  Traffic signs overrule traffic rules, in as far as these rules are incompatible with the signs.

In the following subsection the following abbreviations are used:

- $Q_1(i)$: i is on motorways

- a_p: to drive p km/h, for $p = 0,1,2,\ldots,200$
- b_p : $a_{p+1} \cup a_{p+2} \cup \cdots \cup a_{200}$

[1] Artikel 22 Reglement verkeersregels en verkeerstekens 1990.
[2] Artikel 63 Reglement verkeersregels en verkeerstekens 1990.

- M: The group of motor vehicle drivers.
- E_1: The group of lorry drivers.
- E_2: The group of bus drivers.
- E_3: The group of motor vehicle with a trailer drivers.
- E_4: The group of tractor drivers.
- E_5: The group of self-driving work vehicle drivers.

First a formalisation for *article 22* [3] RVV 1990, which specifies separate speed limits for vehicles, is needed. *Article 22 RVV 1990* is formalised as follows:

$$\text{a. } \forall_{i \in E_1 \cup E_2 \cup E_3} F(i : b_{80})$$

Every individual in the group of lorry drivers, bus drivers and motor vehicle drivers with a trailer is forbidden to drive above 80 km/h.

$$\text{b. } \forall_{i \in E_4 \cup E_5} F(i : b_{25})$$

Every individual in the group of tractor drivers and self-driving work vehicle drivers is forbidden to drive above 25 km/h. For HI's case we need part a. of article 22 since the driver was a lorry driver. *Article 63* [4] of the RVV 1990 states that traffic signs override traffic rules, this is formalised as follows:

$$\forall_{i \in M} F(i : b_{100})$$

Note that this formalisation is already adjusted to HI's case, since the sign on the A28 depicts a maximum speed of 100 km/h. In other cases the formalisation would be different. The formalisation states that: every individual in the group of motor vehicle drivers is forbidden to drive above 100 km/h. With this formalisation we assume that the traffic sign overrules the traffic rules, hence b_{100} . Now let i_1 be the lorry driver from HI's firm. The following formalisation can be done:

$$F(i_1 : b_{80}) \wedge F(i_1 : b_{100})$$

Since $b_{80} \equiv a_{81} \cup a_{82} \cup \cdots \cup a_{200}$ and $b_{100} \equiv a_{101} \cup a_{102} \cup \cdots \cup a_{200}$, the formula is equivalent to:

$$F(i_1 : a_{81} \cup a_{82} \cup \cdots \cup a_{200}) \wedge F(i_1 : a_{101} \cup a_{102} \cup \cdots \cup a_{200})$$

Now using proposition 7 of PD_eL, which can be found in appendix A.1: $\vdash F(\beta_1 \cup \beta_2) \equiv F(\beta_1) \wedge F(\beta_2)$, we can further derive the formula:

[3] Artikel 22 Reglement verkeersregels en verkeerstekens 1990.
[4] Artikel 63 Reglement verkeersregels en verkeerstekens 1990.

$$F(i_1 : a_{81}) \wedge F(i_1 : a_{82}) \wedge \cdots \wedge F(i_1 : a_{200}) \wedge F(i_1 : a_{101}) \wedge F(i_1 : a_{102}) \wedge \cdots \wedge F(i_1 : a_{200})$$

Using proposition 28 of PD_eL, which can also be found in appendix A.1: $\vdash F(\beta) \wedge F(\beta) \equiv F(\beta)$, duplicates can be removed.

$$F(i_1 : a_{81}) \wedge F(i_1 : a_{82} \wedge \cdots \wedge F(i_1 : a_{200})$$

Now using proposition 7 of PD_eL again but now the other way around, we get:

$$F(i_1 : a_{81} \cup i_1 : a_{82} \cup \cdots \cup i_1 : a_{200})$$

Since $b_{80} \equiv a_{81} \cup a_{82} \cup \cdots \cup a_{200}$, the formula can now be formulated as follows:

$$F(i_1 : b_{80})$$

This indicates that the lorry driver is forbidden to drive at a speed above 80km/h, it can be concluded that there is no incompatibility of the traffic sign and the traffic rule. Therefore the ticket was correctly given. The outcome of this model matches the outcome of the original ruling of the judge.

3.2.2 Bourhill v Young

This case has, up to my knowledge, never been worked out in a paper, thus the doctoral thesis by Royakkers [22] is used as a guideline. This case does not have the clear rules HI's case has. In HI's case, the RVV 1990 gives clear "rules" as to what is allowed and what is not. Here however, the requirements are often subjective, making it harder to get a straight formula out of the requirements. Three principles are needed to establish negligence of duty of care: duty of care, breach of duty of care, and proximity.

Furthermore, the neighbour test or neighbour principle from the Donoghue v Stevenson (1932) case can be used to help determine duty of care [8, 5]. Donoghue v Stevenson (1932) is a relatively small case, but the case has had a wide influence in the law. Lord Atkin formulated a general principle (the neighbour principle) in the case, he said:

> "You must take reasonable care to avoid acts or omissions which you can reasonably foresee would be likely to injure your neighbour. Who, then, in law, is my neighbour? The answer seems to be persons who are so closely and directly affected by my act that I ought reasonably to have them in contemplation as being so affected when I am directing my mind to the acts or omissions which are called in question" [5]

As explained in section 2.3, an important principle in the law is Stare Decisis. The neighbour principle is therefore often used in cases that occurred after the Donoghue v Stevenson (1932) case, thus we use it for the Bourhill v Young case which occurred in 1942.

To help decide the case, there are two principles that can be taken from the neighbour principle. These principles need to be the case for duty of care to be true:

1. **Reasonable foresight**: Reasonable foresight means that a person has a degree of foresight of the consequences that his or her actions have on others.

2. **Proximity**: Duty of care is not owned to everybody that you foresee harm too, only those who are closely and directly affected by an action.

In the Bourhill v Young case the two principles can be depicted as follows:

1. When looking at the case facts it is not reasonably foreseeable that Young crashing his motorcycle would harm Mrs. Bourhill who was alighting the tram.

2. Proximity does not necessarily mean physical closeness, but it also means any form of relation between the parties. Here there is no form of relation (family, friends, etc) between Mrs. Bourhill and Young. There is however, 50 feet between them. Thus we can say that it is acceptable to say that Mrs. Bourhill was not sufficiently proximate to the accident.

In 1990, the three-state test from Caparo v Dickman came as an extension to the neighbour principle. Which, since then, is used to determine duty of care [8]. With this test the court has to ask the following three questions to determine duty:

1. Was the risk of injury or harm to the claimant *reasonably foreseeable*?

2. Was there *sufficient proximity* between the parties?

3. Is it *fair, just and reasonable*, on public policy grounds, to impose a duty of care?

If the answer to all these three questions is: yes, duty of care can be established. This can be seen as an elaboration of the neighbour principle. Since the three-state test dates back to 1990 it cannot be used to help determine duty of care in the Bourhill v Young case which occurred in 1942.

In the following subsection these abbreviations are used:

- $\underline{rf_i}$: to have reasonable foresight of harm for i.

- $\underline{br_i}$: to breach the duty of care you have for i.
- p: to be sufficiently proximate.

- M: The group of motor vehicle drivers.
- B: The group bystanders.

Firstly, the principle of duty of care needs to be formalised. The neighbour principle is used to formalise this:

$$\forall_{i\in M}\forall_{j\in B}(O(i:\underline{rf_j}) \wedge {}_iO_j(p))$$

Every motor vehicle driver is obliged to have reasonable foresight of harm to every bystander and it is obligatory for every bystander towards motor vehicle driver to be sufficiently proximate to harm. The motor vehicle driver needs to have reasonable foresight and the bystanders needs to be sufficiently proximate to the harm. Secondly, the principle of breach of duty of care is given by:

$$\forall_{i\in M}\forall_{j\in B}O(i:\underline{br_j})$$

Every motor vehicle driver breaches the duty of care to every bystander. Lastly, the principle of proximity is given by:

$$\forall_{i\in M}\forall_{j\in B\,i}O_j(p)$$

It is obligatory for every bystander towards motor vehicle driver to be sufficiently proximate to harm.

Let i_1 be a motor vehicle driver and j_1 be a bystander. The formalisation for *negligence of duty of care* is as follows:

$$(O(i_1:\underline{rf_{j_1}}) \wedge {}_{i_1}O_{j_1}(p)) \wedge O(i_1:\underline{br_{j_1}}) \wedge {}_{i_1}O_{j_1}(p)$$

Using proposition 12 of PD_eL which can be found in appendix A.1: $\vdash O(\beta_1 \wedge \beta_2) \rightarrow O(\beta_1) \wedge O(\beta_2)$;, this can also be formalised as:

$$O(i_1:\underline{rf_{i_1}}) \wedge {}_{i_1}O_{j_1}(p) \wedge O(i_1:\underline{br_{j_1}}) \wedge {}_{i_1}O_{j_1}(p)$$

Using proposition 29 of PD_eL, which can also be found in appendix A.1: $\vdash O(\beta) \wedge O(\beta) \equiv O(\beta)$, duplicates can be removed.

$$O(i_1:\underline{rf_{j_1}}) \wedge {}_{i_1}O_{j_1}(p) \wedge O(i_1:\underline{br_{j_1}})$$

Now let i_2 be Young and let j_2 be Mrs. Bourhill. As can be seen from the neighbour principle, the two principles of duty of care (reasonable foresight and proximity) cannot be established in the Bourhill v Young case. Since Young did cause an accident, he did breach the duty. Since proximity could not be established for duty of care, it can also not be established for negligence of duty of care because the proximity does not differ. The full formalisation for the *Bourhill v Young* case:

$$(\neg O(i_2 : \underline{rf_{j_2}}) \wedge \neg_{i_2} O_{j_2}(p)) \wedge O(i_2 : \underline{br_{j_2}}) \wedge \neg_{i_2} O_{j_2}(p)$$

Using proposition 12 of PD_eL which can be found in appendix A.1: $\vdash O(\beta_1 \wedge \beta_2) \rightarrow O(\beta_1) \wedge O(\beta_2)$;, this can also be formalised as:

$$\neg O(i_2 : \underline{rf_{j_2}}) \wedge \neg_{i_2} O_{j_2}(p) \wedge O(i_2 : \underline{br_{j_2}}) \wedge \neg_{i_2} O_{j_2}(p)$$

Using proposition 29 of PD_eL, which can also be found in appendix A.1: $\vdash O(\beta) \wedge O(\beta) \equiv O(\beta)$, duplicates can be removed.

$$\neg O(i_2 : \underline{rf_{i_2}}) \wedge \neg_{i_2} O_{j_2}(p) \wedge O(i_2 : \underline{br_{j_2}})$$

This indicates that reasonable foresight and proximity cannot be established, breach of duty can be established. Since the formula of the Bourhill v Young case does not match the formula of negligence of duty of care, negligence of duty of care cannot be established. The outcome of the model for the Bourhill v Young case matches the outcome of the original ruling of the judge.

3.3 Reduction graph representation of the cases

Below, the reduction graph representations of the two cases can be found.

3.3.1 HI's case

This case has, up to my knowledge, never been worked out in a paper, thus the paper by Branting [7], and book by Barr [6] are used as guidelines. Again, before creating the reduction graph, we first need the input. The formula for the ticket for the driver was already shown before, now shown for reference:

ticket for the driver $\leftrightarrow$ $\neg$traffic rules override signs $\wedge$ the driver is a lorry driver $\wedge$ lorries cannot drive faster than 80 km/h $\wedge$ the driver drove 81 km/h or more

The following facts are known:

- The driver from HI's firm is a lorry driver.
- The driver drove 96 km/h on the A28.
- The traffic sign on the A28 depicts 100 km/h.

The following rules come from legal authorities, already shown in section 3.2.1, they are now shown for reference:

- **Article 22 RVV 1990**[5]:
  Voor zover niet ingevolge andere artikelen een lagere maximumsnelheid geldt, gelden voor de volgende voertuigen de volgende bijzondere maximumsnelheden:

[5] Artikel 22 Reglement verkeersregels en verkeerstekens 1990.

a. voor vrachtauto's, autobussen en motorvoertuigen met aanhangwagen 80 km per uur;

b. voor tractoren en zelfrijdende werktuigen 25 km per uur.

Which translates to:
As far as in other articles a lower speed limit is not in order, the following maximum speed limits apply to vehicles:

a. for lorry's, busses and motor vehicles with a trailer 80 km per hour;

b. for tractors and self-driving work vehicles 25 km per hour.

- **Article 63 RVV 1990**[6]:
  Verkeerstekens gaan boven verkeersregels, voor zover deze regels onverenigbaar zijn met deze tekens.
  Which translates to:
  Traffic signs overrule traffic rules, in as far as these rules are incompatible with the signs.

In this case there is an evaluation criterion needed, this because in article 63 RVV 1990[7] it states: in as far as these rules are incompatible with the signs. This can be read with several interpretations:

- When there is a sign, the sign overrules the traffic rules [22].
- When the sign states something it is only the case when it is in conflict with the rule.

The criterion used in this thesis: When there is a possibility to adhere to both the traffic rules and sign this needs to be done.

Two reduction graphs have to be made, one where *ticket for the driver* is the initial problem, the other where $\neg$*ticket for the driver* is the initial problem. The full scale images of the reduction graphs can be found in appendix section: A.2.

Firstly the reduction graph where the *ticket for the driver* is the initial problem. This reduction graph can be seen in figure 3.1. "Ticket for the driver" is the initial problem, it is also an AND-node. It has four subproblems that each need to be solved. As can be seen in the reduction graph all four nodes end as a terminal node. The graph is therefore solvable.

Secondly the reduction graph where the $\neg$*ticket for the driver* is the initial problem. This reduction graph can be seen in figure 3.2. Like in the Bourhill v Young case, the formula needs to be changed since there is a different interpretation:

[6] Artikel 63 Reglement verkeersregels en verkeerstekens 1990.
[7] Artikel 63 Reglement verkeersregels en verkeerstekens 1990.

$$\neg\text{ticket for the driver} \leftrightarrow \text{when there is a sign, it overrules the traffic rules} \wedge \text{the driver is a lorry driver} \wedge \text{lorries cannot drive faster than 80 km/h} \wedge \text{the driver drove 81 km/h or more}$$

Here ¬*ticket for the driver* is also an AND-node. It still has four sub-problems to be solved. Since the law in article 63 RVV 1990 is ambiguous, here also the four sub-problems end in terminal nodes. Thus this reduction graph is also solvable.

Above we gave an evaluation criterion: when there is a possibility to adhere to both the traffic rules and sign this needs to be done. Now the two reduction graphs can be evaluated using this evaluation criterion. For HI's case, the first reduction graph (figure 3.1) will be ranked higher because the sign states that the speed limit is 100 km/h but not that every vehicle needs to drive 100 km/h. Therefore the traffic rule and traffic sign are not necessarily incompatible. The outcome of the model matches the outcome of the original case.

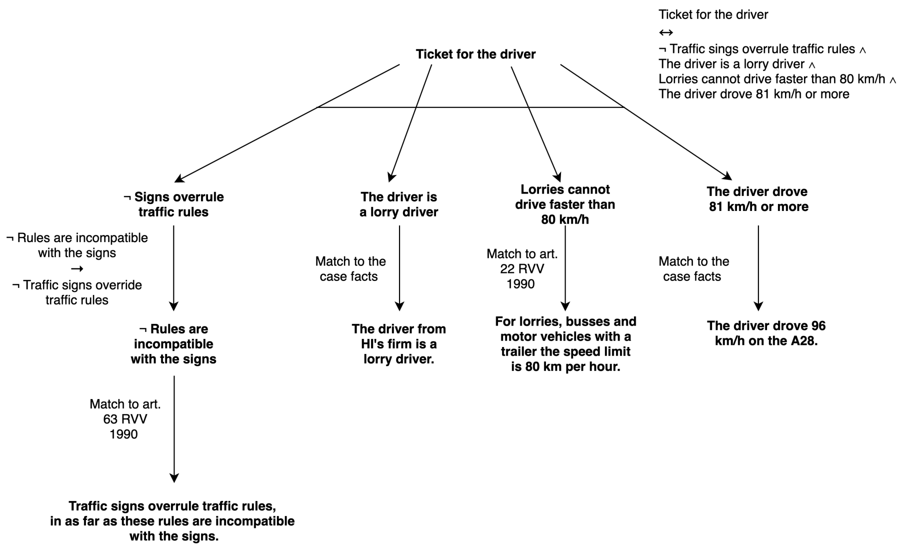

Figure 3.1: Reduction-graph representation of HI's case with ticket for the driver as the initial node.

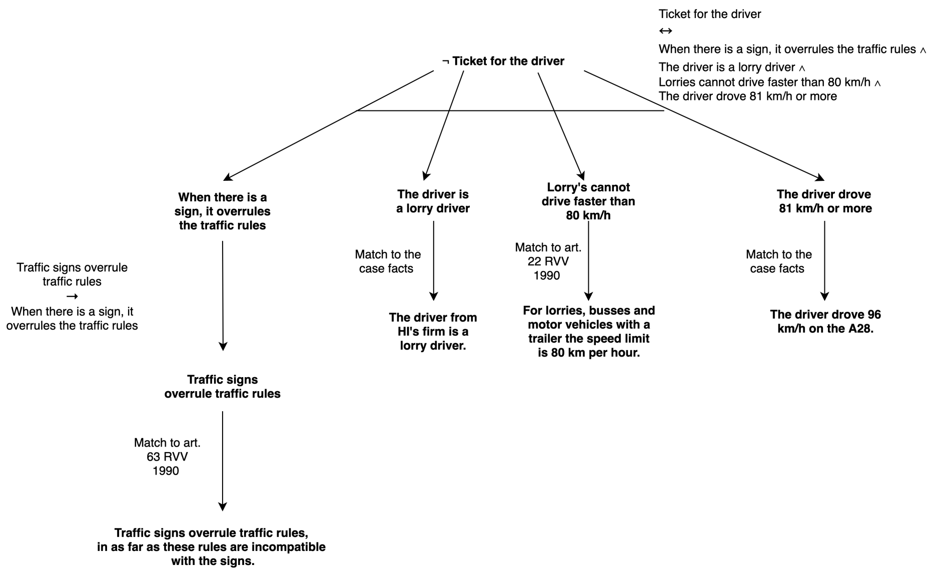

Figure 3.2: Reduction-graph representation of HI's case with ¬ ticket for the driver as the initial node.

3.3.2 Bourhill v Young

This case was originally worked out in the paper by Branting [7]. The worked out case is used as a guideline. Before creating the reduction graph, we first need the input. We want to know whether Young has neglected his duty of care or not. The formula for negligence of duty of care was already shown before, now shown again for reference:

$$\text{negligence of duty of care} \leftrightarrow \text{duty} \wedge \text{breach of duty} \wedge \text{proximate cause}$$

The following facts are known:

- Young caused an accident when a tram was stopped near by.
- The tram was stopped about 50 feet before the accident.
- Mrs. Bourhill was alighting from the stopped tram.
- Mrs. Bourhill suffered a nervous shock from seeing blood on the road.

The neighbour test or neighbour principle from the Donoghue v Stevenson (1932) case, discussed in section 3.2.2 is again used to help determine duty of care [8, 5]. For reference the two principles and their application to the Bourhill v Young case are shown again. These principles need to be the case for duty of care to be true:

1. **Reasonable foresight**: Reasonable foresight means that a person has a degree of foresight of the consequences that his or her actions have on others.

2. **Proximity**: Duty of care is not owned to everybody that you foresee harm too, only those who are closely and directly affected by an action.

In the Bourhill v Young case the two principles can be depicted as follows:

1. When looking at the case facts it is not reasonably foreseeable that Young crashing his motorcycle would harm Mrs. Bourhill who was alighting the tram.

2. Proximity does not necessarily mean physical closeness, but it also means any form of relation between the parties. Here there is no form of relation (family, friends, etc) between Mrs. Bourhill and Young. There is however, 50 feet between them. Thus we can say that it is acceptable to say that Mrs. Bourhill was not sufficiently proximate to the accident.

The neighbour principle can be put into the following formula:

$$\text{duty of care} \leftrightarrow \text{reasonable foresight} \wedge \text{proximity}$$

In this case there are no arguments stronger or weaker, thus an evaluation criterion is not needed.

The initial problem that has to be solved is: did Young neglect his duty of care. Therefore two reduction graphs are made: one where the initial problem is *neglect of duty of care*, and the other where the initial problem is ¬*neglect of duty of care*. The full scale images of the reduction graphs can be found in appendix section: A.2.

First, the reduction graph when the initial problem is *neglect of duty of care*. This reduction graph can be seen in figure 3.3. Firstly, the formula of *neglect of duty of care* is followed, which creates three sub-problems: *duty*, *breach of duty* and *proximate cause*. *Neglect of duty of care* is an AND-node and all sub-problems need to be solved. *Duty* is also an AND-node, its sub-problems are *Reasonable foresight* and *Proximity*. As can be seen above, these are non-terminal nodes with no descendants and thus cannot be solved. We thus cannot solve the sub-problem and ultimately we cannot solve the initial problem. The same goes for *Proximate cause*, which also is a non-terminal node that has no descendants. The only solvable node is *Breach of duty*, because Young did cause a loud accident. Since the initial node is an AND-node, this reduction graph is not solvable.

Secondly the reduction graph when the initial problem is ¬*Neglect of duty of care*. Since now the negation of the previous used neglect of duty of care is used, the formula is a bit different:

$$\neg\text{neglect of duty of care} \leftrightarrow \neg\text{duty} \vee \neg\text{breach of duty} \vee \neg\text{proximate cause}$$

This is because for ç to be true, duty <u>and</u> breach of duty <u>and</u> proximate cause need to be true. Thus if the negation of neglect of duty of care is true, duty <u>or</u> breach of duty <u>or</u> proximate cause needs not to be true. This makes the initial node an OR-node. The same goes for the *duty* sub-problem. This formula can be changed to:

$$\neg\text{duty} \leftrightarrow \neg\text{reasonable foresight} \vee \neg\text{proximity}$$

For the sake of completeness, the full reduction graph is shown. Since the initial node is an OR-node only one of the arcs needs to be solved for the graph to be solvable. The reduction graph is shown in figure 3.4. It can be seen that in this version of the reduction graph two of the three subproblems of ¬*neglect of duty of care* are terminal nodes and thus are solved. Because the initial node is an OR-node, this makes it a solvable graph.

This outcome of this model matches the outcome of the original case.

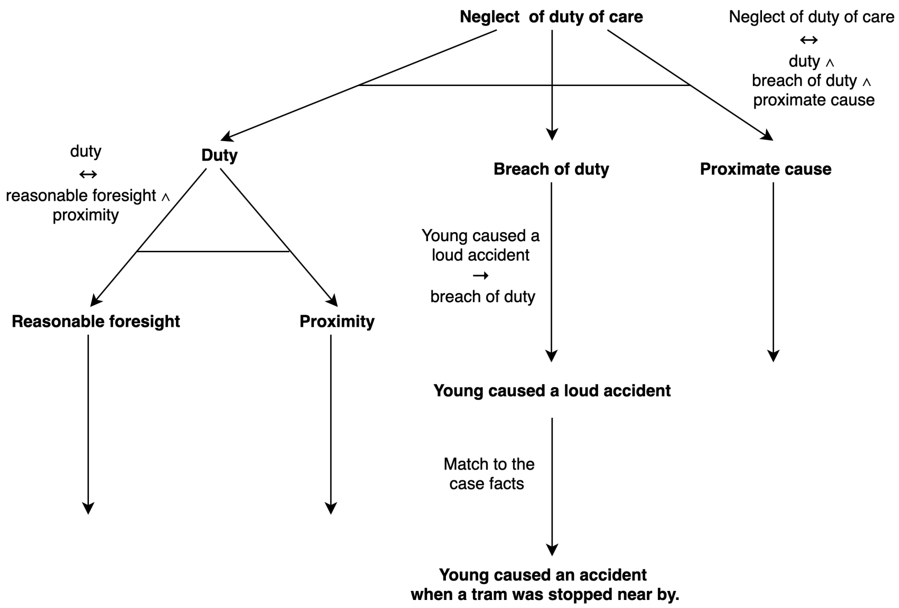

Figure 3.3: Reduction-graph representation of Bourhill v Young with *Neglect of duty of care* as the initial node.

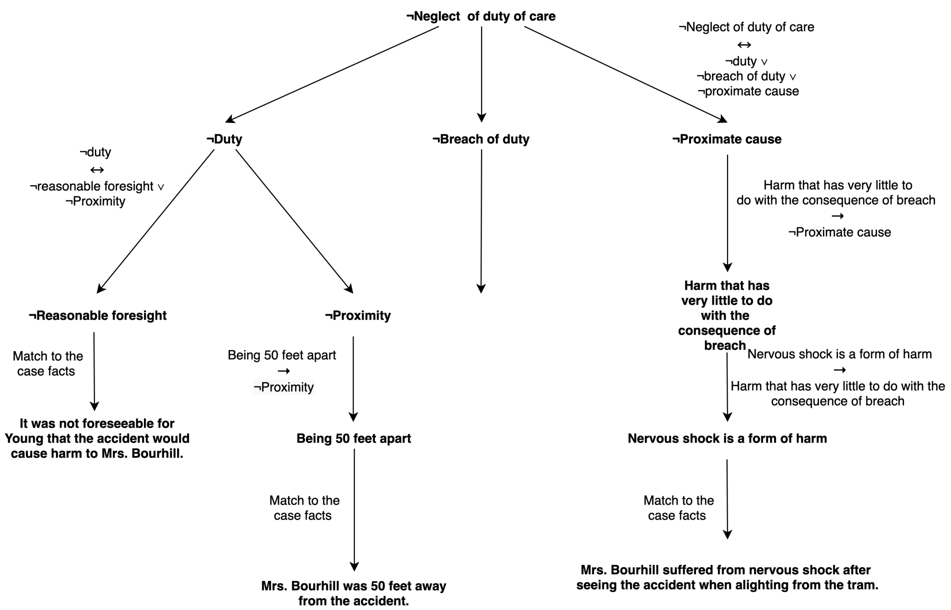

Figure 3.4: Reduction-graph representation of Bourhill v Young with ¬ *Neglect of duty of care* as the initial node.

Chapter 4

Discussion of the Logical Models

To compare the two logical models, multiple criteria are needed. The models will be compared on the following criteria :

- **Correctness of the judgement**: This means asking the question, is the outcome of the models the same as the judge originally ruled?
- **Complexity of the model**: How complex it is to get to the final model.
- **Readability of the model**: How easy it is to understand the model with limited knowledge of the logic behind the model.
- **Usability in law**: Is the model usable in more cases than the cases used in this thesis?

Correctness of the judgement: With all the models presented above the outcomes of the model were equal to the judgement of the judge. This, however, raises the question if the judgement of the case is the final judgement. In America around 20 percent of the cases are appealed [10]. An appealed case is a case that will be reviewed by a higher court in the hope that the decision made by a lower court might be reversed or changed [2]. In this thesis, for both the Bourhill v Young case and HI's case, the judgement of the original case is used to compare the outcome of the model to the judgement. However, in HI's case there was an appeal to the court of cassation. The High Court (Hoge Raad) issued the following statement: "Verweer is terecht en op goede gronden verworpen"[1] (The defence is rejected justly and on good grounds), thus not reversing or changing the original judgement. To my knowledge, there was no appeal in the Bourhill v Young case. Thus it

[1]DD 94.137 [1]

is assumed that the judgements used in this thesis are the final judgements, and thus the outcome of the models matches the outcome of the cases.

Complexity of the model: Both deontic logic and the reduction graph representation have a basis in propositional logic, they both use this in a very different manner. Deontic logic adds an axiom to the logic **K** to get to the logic **D**. From the logic **D**, SDL and PD_eL are added, these additions are quite complex and need a relatively large time investment if one is not familiar with these logics. For the reduction graph representation on the other hand, the only addition made is the AND/OR nodes that need to be understood. Once the rules of the AND/OR graph are known, it is relatively easy to construct an reduction graph from the formula of the claim and the case facts. This deontic model, compared to the reduction graph representation, can thus seen as a relatively complex model.

Readability of the model: Deontic logic and the reduction graph representation have a very different way of showing the case facts and the outcome of the model. The deontic model shows the outcome of the model in the form of a logical formula. The reduction graph representation shows the outcome of the model in several reduction graphs, ranked on criteria if needed. The reduction graph representation provides a rather visual outcome. When having limited knowledge of logic, often a visual representation can make it easier to comprehend the logic used. When a model is made for a legal case, it most of the time has to be interpreted by a jurist. Since the reduction graph representation has both the visual aspect and less prerequisite logical knowledge than the deontic model, this would be easier to use for someone with a limited logical knowledge.

Usability in law: Both models are not a self-contained manner of working with legal cases, they still need to be presented to a jurist for a review. This mostly has to do with the complexity of law. As said before, the law is not only about strict rules, but also about "common sense". The deontic logic approach is, however, more suited for cases involving rules and laws that do not need a subjective judgement. As can be seen in HI's case, the case is mostly about following the rules of the RVV 1990, which works very well in deontic logic. In the Bourhill v Young case a more subjective approach is needed since there are no strict boundaries on what for example reasonable foresight and proximity are. In deontic logic, the Bourhill v Young case was therefore not as straightforward to model as HI's case. The reduction graph representation has more room for subjective reasoning. Therefore using a reduction graph representation it is relatively easier to model an case where a more subjective reasoning is needed. Both deontic logic and the reduction graph representation can be used in a wide variety of cases. When looking at the cases in this thesis, deontic logic is more suited for cases with relatively strict rules and the reduction graph representation is suited for both cases using relatively strict rules and cases needing a subjective approach.

Chapter 5

Conclusions

This thesis aimed to answer the question whether modal logic is an advantage over not using modal logic in logical models for solving legal problems. To get a better insight in this problem, two separate legal cases were used: the Bourhill v Young case and HI's case. To answer the research question, the logics used in this thesis needed to be introduced. Both deontic logic and the reduction graph representation were elaborated on to the extent needed for the two cases. After introducing deontic logic and reduction graph representation, the models were created using both these logics.

After comparing the models, it became clear that both deontic logic and the reduction graph representation were suited for solving the two cases used in this thesis. Both models had outcomes that matched the judgement of the original cases. Taking also the complexity of the model and the usability in the law into account, the reduction graph representation worked, relatively and based on the two cases used, more intuitively and therefore better. Since this thesis only covers a small portion of the logic used in law, there is no general answer to the research question. What can be said is that, when representing legal cases similar to the Bourhill v Young case and HI's case, using deontic logic is not an advantage over using the reduction graph representation due to the complexity and readability of the deotic logic models.

5.1 Future work

This thesis only shows a tip of the iceberg on both logic in law but also deontic logic and reduction graphs in law. There are many possibilities to further expand this research.

1. Firstly, within this thesis the models are purely made by hand. For both the deontic logic and the reduction graph representation, it would be an advantage to implement them in a knowledge based system.

Branting already showed in his paper that it is possible to implement a reduction graph in GREBE [7]. Royakkers also suggested that it would be possible to implement the theories of both SDL and PD_eL in a knowledge based system [22].

- Royakkers already discussed this in the adaptation of his doctoral thesis: Beth-tableau's can be used to check if a certain formula is valid [23]. This could then also be incorporated in the knowledge based systems.
- Branting used GREBE in his paper to show that the reduction graph representation could be implemented [7]. Since the visual representation proved helpful, the GREBE system could also be expanded by implementing such a visual representation that is used within this thesis.

2. Secondly, within this thesis only two legal cases are made in both models. This does not give an accurate representation of how both models can be used in a wide variety of cases. Therefore, it would be good if the models are worked out for more cases with a different origin.

    - Within this thesis, the outcome of the case was already known when creating the models. This causes a slight bias when constructing the model. This can be prevented by constructing the models without knowing the outcome of the case beforehand.

3. Lastly, as said before, this thesis only covered a small portion of the possible logical modals that could be used within the law. Therefore more logical modals already used within the law can be explored further. Looking at expanding logical models currently not used in the law to suit the law is also worth looking at. It might provide new insights to how logic is used in the law.

Bibliography

[1] *94.I37 (61-91-V)*, Delikt & Delinkwent **24** (1994), no. 3, 332–333.

[2] *What is an appeal?*, https://www.courtofappealbc.ca/appellant-guidebook/1.1-what-is-an-appeal, apr 2020.

[3] Anonymous, *Square of Opposition*, https://www.iep.utm.edu/sqr-opp/.

[4] Roberta Ballarin, *Modern Origins of Modal Logic*, https://plato.stanford.edu/entries/logic-modal-origins/, may 2017.

[5] Kit Barker, *Who is my neighbour? Legal reasoning and Lord Atkin*, https://www.youtube.com/watch?v=9v9JO5sqTp4, nov 2017.

[6] Avron Barr and Edward A. Feigenbaum, *The handbook of Artificial Intelligence Volume 1*, Addison-Wesley Publishing Company Inc., 1949.

[7] L Karl Branting, *A reduction-graph model of precedent in legal analysis*, Artificial Intelligence **150** (2003), 59–95.

[8] CILEx Law School Ltd, *The Duty of Care*, 2018, pp. 9–12.

[9] Jyoti Dabass and Bhupender Singh Dabass, *Scope of Artificial Intelligence in Law*, (2018).

[10] Theodore Eisenberg, *Appeal Rates and Outcomes in Tried and Non-tried Cases: Further Exploration of Anti-Plaintiff Appellate Outcomes*, Journal of Emperical Legal Studies **1** (2004), no. 3, 659–688.

[11] James Garson, *Modal logic*, https://plato.stanford.edu/entries/logic-modal/#Rel, 2018, pp. 1–22.

[12] Davide Grossi and Antonino Rotolo, *Logic in the Law: A Concise Overview*, Logic and Philosophy Today. College Publications (2011).

[13] S. Hamad, *Creativity: Method or Magic?*, Consciousness and Cognition (2007), 127–137.

[14] Henning Herrestad, *Is modal logic necessary or to be desired for expert systems in law?*, https://www.jus.uio.no/ifp/om/organisasjon/afin/forskning/notatserien/1996/2_96_b.html, 1996.

[15] Gerald N. Hill and Kathleen Hill, *The people's law dictionary: Taking the mystery out of legal language*, MFJ Books, 2002.

[16] Ken LaMance, *What is Battery? — LegalMatch*, https://www.legalmatch.com/law-library/article/what-is-battery.html, jun 2018.

[17] Law Insider, *Definition of Legal Authority*, https://www.lawinsider.com/dictionary/legal-authority.

[18] LawTeacher, *Bourhill v Young [1943] AC 92*, https://www.lawteacher.net/cases/bourhill-v-young.php, nov 2013.

[19] Paul McNamara, *Deontic Logic*, sep 2017, pp. 1–22.

[20] Oracle, *Atomic Access*, https://docs.oracle.com/javase/tutorial/essential/concurrency/atomic.ht

[21] Terence Parsons, *The Traditional Square of Opposition*, https://plato.stanford.edu/entries/square/, apr 2017.

[22] Lambèr Maarten Maria Royakkers, *Representing Legal Rules ln Deontic Logic*, Tech. report, University of Tilburg, Tilburg, 1996.

[23] Lamber M.M. Royakkers, *Extending Deontic Logic for the Formalisation of Legal Rules*, Springer Science+Business Media Dordrecht, 1998.

[24] L.M.M. Royakkers, *Kunnen maximumsnelheden conflicteren?* , ANWB Verkeersrecht (1995), 39.

[25] Arend Soeteman, *Normative Judgements*, pp. 23–46, Springer Netherlands, Dordrecht, 1989.

[26] Lawrence Solum, *Legal Theory Blog: Normative Legal Theory*, https://lsolum.typepad.com/legaltheory/normative_legal_theory/, 2009.

[27] Harry Surden, *Artificial Intelligence and Law: An Overview*, Tech. report, 2019.

[28] Stephen E. Toulmin, *The Uses of Argument, Updated Edition*, Cambridge University Press, 2003.

[29] G H von Wright, *Is and Ought*, pp. 31–48, Springer Netherlands, Dordrecht, 1986.

[30] G.H. Von Wright, *Deontic Logic*, Mind **60** (1951), no. 237, 1–15.

[31] R.J. Wieringa and J.-J.H. Meyer, *Applications of Deontic Logic in Computer Science: A Concise Overview*, Deontic Logic in Computer Science: Normative System Specification (Amsterdam) (John-Jules Ch. Meyer and Roelf J. Wieringa, eds.), Wiley, 1993, pp. 17–40.

[32] Julie Young, *Stare Decisis Definition*, https://www.investopedia.com/terms/s/stare_decisis.asp, 2019.

Appendix A

Appendix

A.1 Propositions of PD_eL

1. $\vdash [\beta]\, true$
2. $\vdash [\beta]\, (\Phi_1 \wedge \Phi_2) \equiv [\beta]\, \Phi_1 \wedge [\beta]\, \Phi_2$;
3. $\vdash [\beta]\, \Phi_1 \vee [\beta]\, \Phi_2 \rightarrow [\beta]\, (\Phi_1 \vee \Phi_2)$;
4. $\vdash [\beta_l]\, \Phi_1 \wedge [\beta_2]\, \Phi_2 \rightarrow [\beta_1 \wedge \beta_2]\, (\Phi 1 \wedge \Phi 2)$;
5. $\vdash [\beta_1]\, \Phi 1 \wedge [\beta_2]\, \Phi 2 \rightarrow [\beta 1 \cup \beta_2]\, (\Phi 1 \vee \Phi 2)$;
6. $\vdash [\beta_1 \cup \beta_2 \cup (\beta_1 \wedge \beta_2)]\, \Phi \equiv [\beta_1 \cup \beta_2]\, \Phi$;
7. $\vdash F(\beta_1 \cup \beta_2) \equiv F(\beta_1) \wedge F(\beta_2)$;
8. $\vdash F(\beta_1) \vee F(\beta_2) \rightarrow F(\beta_1 \wedge \beta_2)$;
9. $\vdash F(\beta_1; \beta_2) \equiv [\beta_1]\, F(\beta_2)$;
10. $\vdash O(\beta_1; \beta_2) \equiv O(\beta_1) \wedge [\beta_1]\, O(\beta_2)$;
11. $\vdash O(\beta_1) \vee O(\beta_2) \rightarrow O(\beta_1 \cup \beta_2)$;
12. $\vdash O(\beta_1 \wedge \beta_2) \rightarrow O(\beta_1) \wedge O(\beta_2)$;
13. $\vdash O(\beta_1) \wedge O(\beta_2) \rightarrow O(\beta_1 \wedge \beta_2)$;
14. $\vdash P(\beta_1 \cup \beta_2) \equiv P(\beta_1) \vee P(\beta_2)$;
15. $\vdash P(\beta_1 \wedge \beta_2) \rightarrow P(\beta_1) \wedge P(\beta_2)$;
16. $\vdash P(\beta_1; \beta_2) \equiv < \beta_1 > P(\beta_2)$;
17. $\vdash F(\beta) \equiv < \beta > \neg V$;
18. $\vdash P(\beta) \equiv \neg O(\overline{\beta})$;

19. $\vdash F(\beta_1) \to F(\beta_1 \wedge \beta_2)$;

20. $\vdash O(\beta_1 \wedge \beta_2) \to O(\beta_1)$;

21. $\vdash O(\beta_1) \to O(\beta_1 \cup \beta_2)$;

22. $\vdash O(\beta_1 \cup \beta_2) \wedge F(\beta_1) \to O(\beta_2)$;

23. $\vdash F(\beta_1 \wedge \beta_2) \wedge O(\beta_1) \equiv F(\beta_2) \wedge O(\beta_1)$;

24. $\vdash F(\beta_1 \wedge (\beta_1; \beta_2)) \equiv F(\beta_1; \beta_2)$;

25. $\vdash F(\beta_1 \cup (\overline{\beta_1} \wedge \beta_2)) \equiv F(\beta_1) \wedge F(\beta_2)$;

26. $\vdash O(\beta_1 \wedge \beta) \equiv O(\beta_1)$;

27. $\vdash O(\beta_1 \cup \overline{\beta_1}) \equiv [fail]V \equiv true$

28. $\vdash F(\beta) \wedge F(\beta) \equiv F(\beta)$

29. $\vdash O(\beta) \wedge O(\beta) \equiv O(\beta)$

30. $\vdash P(\beta) \wedge P(\beta) \equiv P(\beta)$

A.2 Reduction graph representations full scale

Below the full scale figures of the Bourhill v Young case and HI's case can be found.

1. Figure A.1 is the reduction graph of Bourhill v Young with n*eglect of duty of care* as initial node;
2. Figure A.2 is the reduction graph of Bourhill v Young with ¬*neglect of duty of care* as initial node;
3. Figure A.3 is the reduction graph of HI's case with *ticket for the driver* as initial node;
4. Figure A.4 is the reduction graph of HI's case with ¬*ticket for the driver* as initial node.

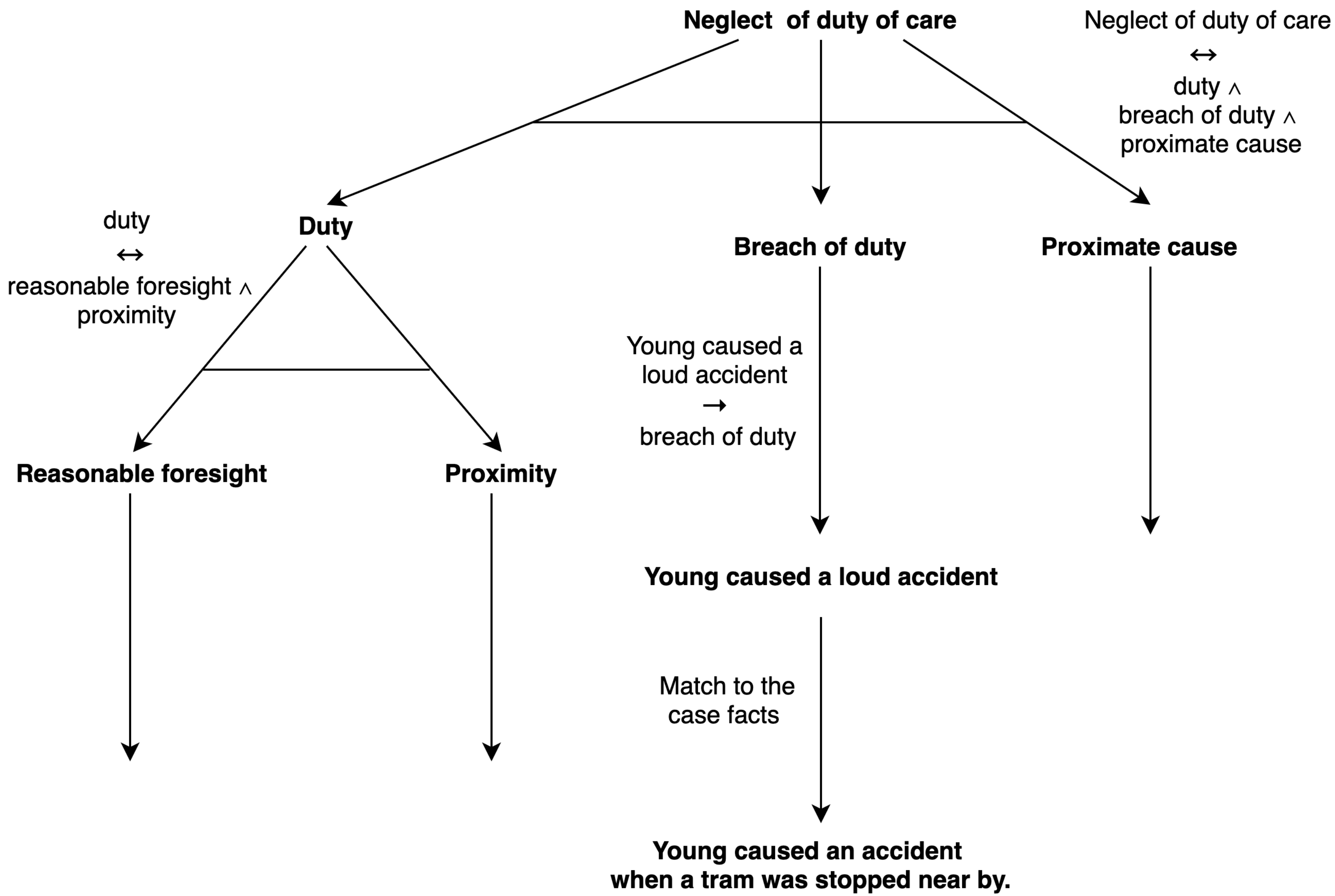

Figure A.1: Reduction graph representation of Bourhill v Young with neglect of duty of care as the initial node

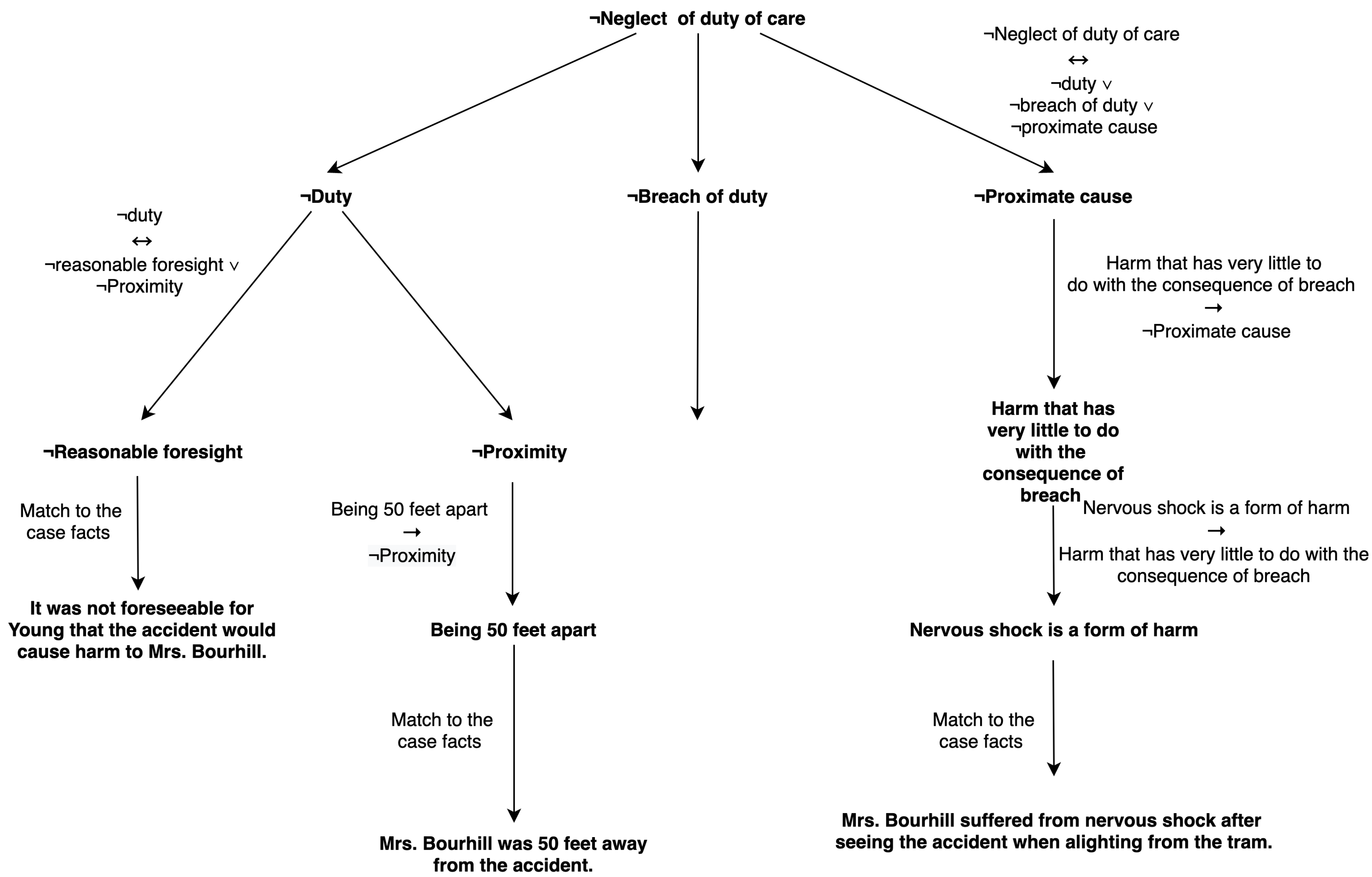

Figure A.2: Reduction graph representation of Bourhill v Young with ¬ neglect of duty of care as the initial node

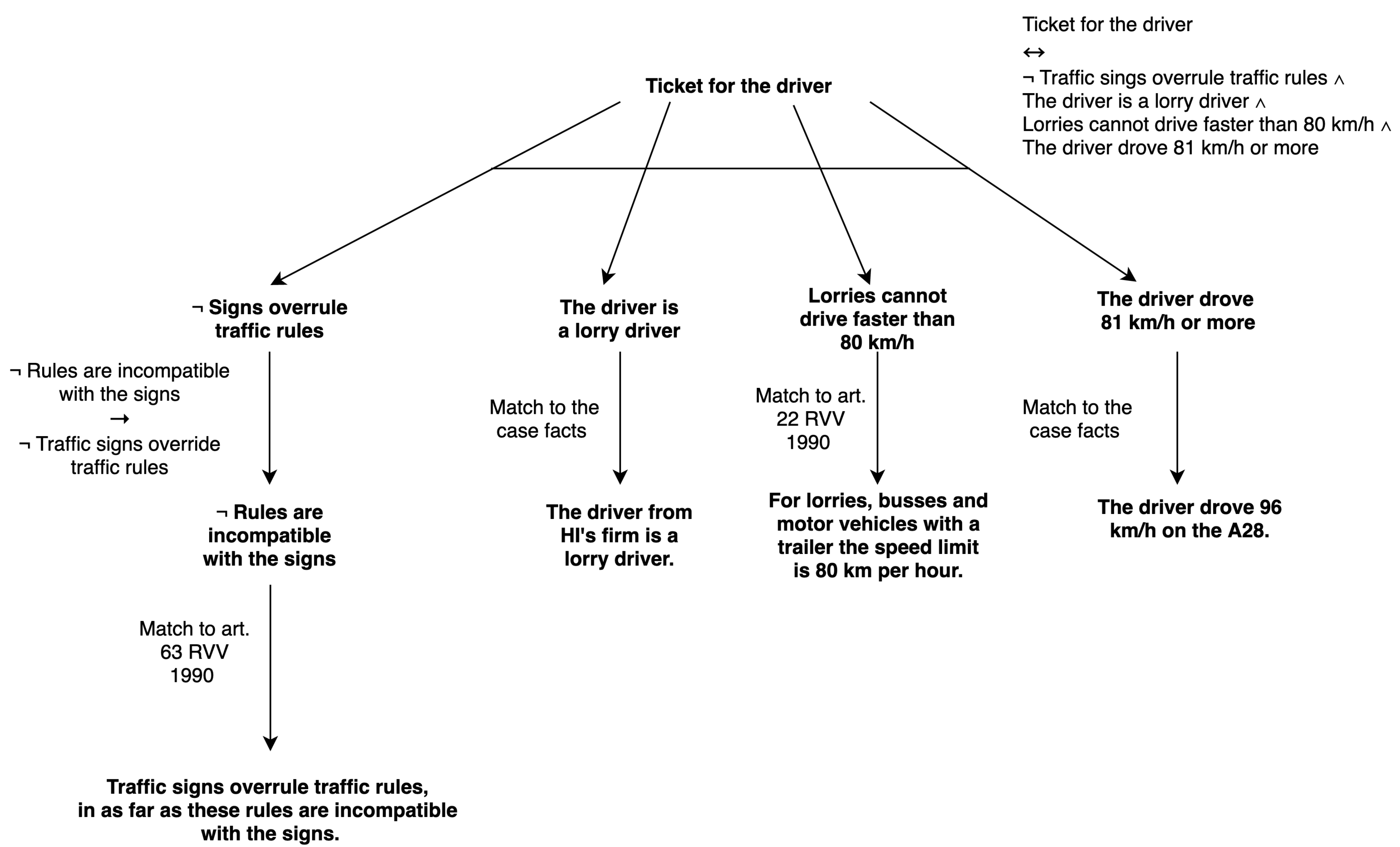

Figure A.3: Reduction graph representation of HI's case with ticket for the driver as the initial node

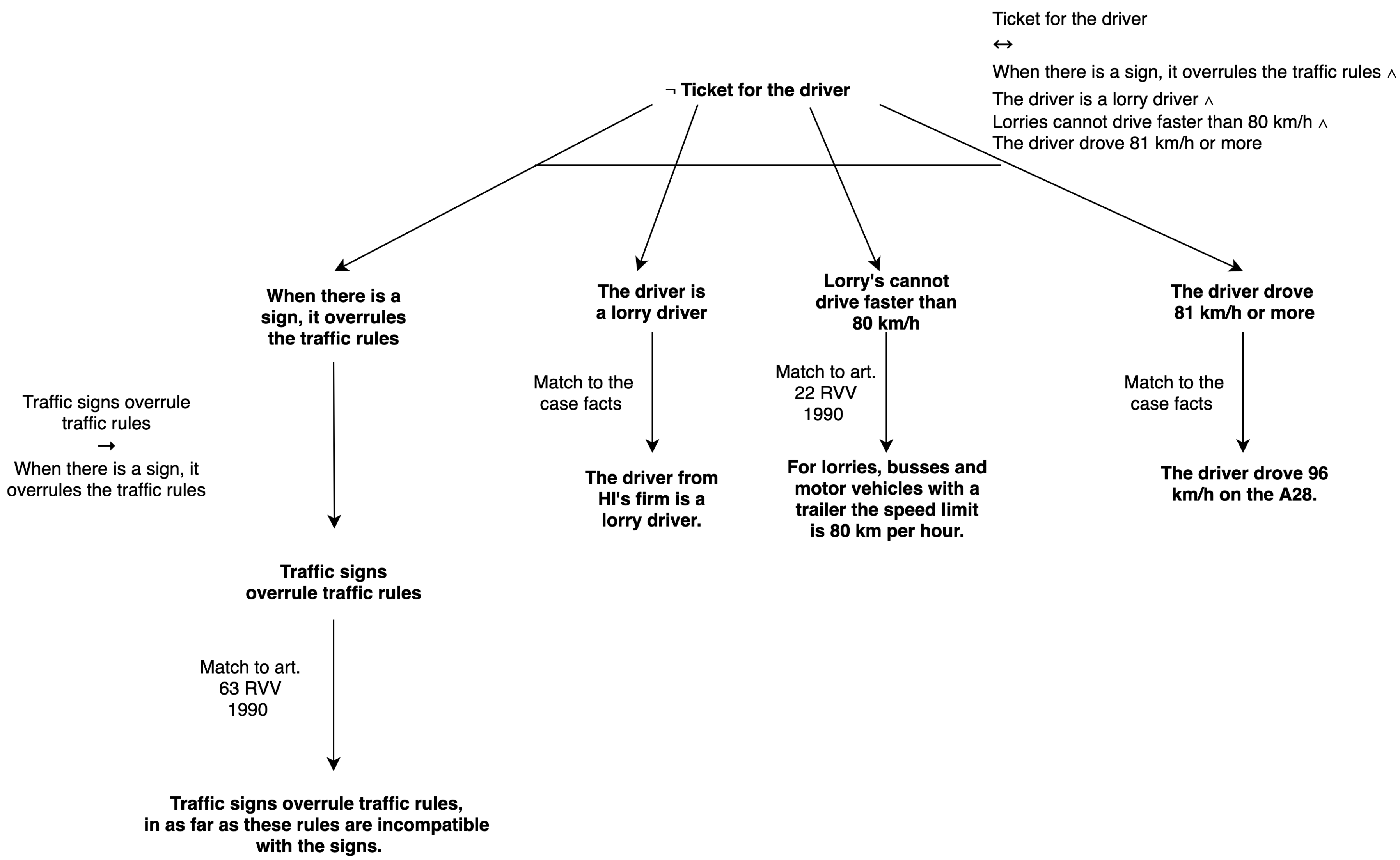

Figure A.4: Reduction graph representation of HI's case with ¬ ticket for the driver as the initial node