A dialogue framework for processing police reports of online trade fraud

Jesse P.S. Six Dijkstra

October 31, 2018

MSc Thesis Artificial Intelligence

Supervisor: Prof. dr. F.J. Bex

Utrecht University

Contents

1 Introduction

Abstract

In this thesis project, an overarching framework for generating dialogues concerning the intake and subsequent case filing of police reports, is constructed. The framework consists of two separate dialogue systems, both with different participant roles and protocols. By applying argumentation rules for online trade fraud to these systems, dialogues between an artificial agent and supposed victims of this type of crime can be generated, such that the submitted police reports can be worked through, relevant follow-up questions can be asked and an investigation can be initiated based on the acquired information. The end result is a structured argument that describes a clear narrative of the situation and sketches a profile of the suspect(s) and their modus operandi.

Online trade is the buying and selling of goods and services through an online platform, such as a webshop, a trade website or social media. Someone is a victim of online trade fraud (or e-fraud) if they have intentionally been swindled by their counterparty in such a transaction. In the Netherlands, alleged victims can report e-fraud through an online intake system on the official police website. We will use the term 'applicant' to denote someone who has submitted a police report. In order to handle an overload of reports concerning e-fraud complaints, the Dutch National Police aims to automatize a large part of the intake process. In collaboration with Utrecht University, a multi-agent system that can classify online complaints as to whether or not they seem to concern e-fraud is currently being developed for the project 'Intelligence Amplification for Cybercrime' (IAC) [3].

The process of e-fraud report intake and its follow-up investigation can roughly be divided into 8 parts:

1. Interpreting the textual input of a police report.
2. Determining if any crucial data is missing.
3. Interacting with the applicant to complete the missing information, if necessary.
4. Making a judgment call on whether the conflict presumably concerns fraud.
5. Matching multiple reports to a single perpetrator.
6. Investigating the leads provided by the reports in order to identify the suspect.
7. Constructing an argument describing the alleged identity and modus operandi of the suspect.
8. Reporting this argument in natural language in the form of a case file.

After step 8, it is up to the Dutch prosecutor's office (*Openbaar Ministerie*) to decide if prosecution is sensible. We can describe steps 1-4 as the *intake process* and steps 5-8 as the *case file process*. Currently, these two processes are executed by different people at separate locations, as the case file process is specifically the responsibility of a subdivision of the Dutch National Police, namely the LMIO (the national contact point for internet fraud). In this thesis project, we aim to bridge these two processes by finding a way to preserve the information acquired in the intake process in a way that it can subsequently be used directly for the case file process, without human intervention necessary. We have decided to approach this from a dialogue-based perspective, which we will justify in the following paragraphs.

There have been attempts to use machine learning on the raw police reports by Kos, Schraagen, Brinkhuis & Bex [6]. They applied algorithms such as multinomial Naive Bayes and logistic regression to a Dutch e-fraud report corpus to predict whether a report will be withdrawn, thus short cutting the intake process by skipping directly to step 4. However, the techniques could not accomplish classification on a satisfactory level (F1-score = 0.594). Later research by van 't Hul [17] managed to significantly surpass these results (F1-score = 0.808). Still, even a binary machine learning algorithm that would perform at 100% accuracy predicts little of substance about a police report beyond its likelihood of withdrawal, which does not give applicable input representation for the case file process.

A useful way to accomplish an unambiguous representation of input and output is through formal argumentation logic. By treating all statements describing aspects of the case as propositions, and by constructing arguments out of these propositions, we have an intuitive and explicable method of reasoning about legal situations. An argument that is made at some point during either process could afterwards be used as a justification for the prosecution of the suspect. Furthermore,

argumentation logic is non-monotonic, as defeasible inferences can be made and earlier conclusions can be withdrawn based on new information. Criminal cases are often at least partially based on abductive logic and presumptions, due to the fact that getting a complete picture of a conflict is practically inachievable. If crucial information is revealed at a later stage, this could overturn earlier conclusions.

Some type of natural language processing (NLP) is required to interpret the content of submitted reports. The main stumbling block for the utilization of NLP on Dutch e-fraud reports so far has been the fact that the applicable resources, including the number of relevant corpora, are very limited. Named Entity Recognition (NER) has been tested on online Dutch fraud reports, using the token set SoNaR-1 from the NER module in the free natural language processing suite Frog [16]. The performance of the current algorithm is not sufficient for online criminal complaints, due to several reasons. Because of its stylistically and grammatically low quality, the intake format is not optimally suited for supervised machine learning. Also, the assumption that a token entity occurs in existing databases and web corpora does not hold for private data. Furthermore, active learning and crowd sourcing is not widely applicable to the data set due to legal constraints. Although the Dutch National Police is currently working on creating an expansive corpus of relevant entities, the linguistic interpretation of e-fraud intake is still very much a work in progress. It is, however, not the goal of this project to solve this problem. For the inter-agent dialogue framework we will assume that all involved parties can interpret each others locutions fully. Given that all locutions are represented as propositions, we can reason forward from step 2 of the intake process.

By modelling the interaction between agents in the system as an argumentation dialogue, we get the advantages of argumentation structures mentioned earlier, with the additional benefit of allowing parties to respond to each other's arguments. If any unclarity were to exist about the statements made by one of the parties, another party could challenge it and thus ask that they explain their reasoning. If a party were to assert something that is contradictory to information provided earlier, then any weak spots in their argument could be attacked on that basis.

In this thesis project, we aim to answer the following four research questions:

1. Can we construct a dialogue system for intake dialogues through which the validity of fraud claims is established and missing information can be requested?
2. Can we construct a dialogue system for case file dialogue, that combines information from police reports and external sources in order to formulate a coherent argument?
3. Can we combine these two overarching dialogue systems into a single formal framework?
4. Can we structure the resulting arguments in a way that they can be transcribed into natural language?

In doing so, we attempt to construct a dialogue framework for generating dialogues that model the intake and case file processes. This framework has a generic set-up, such that theoretically it could be applied to reports of crimes of any nature given the right argumentation rules, but the implementation described in this paper is online trade fraud specific. This thesis has the following set-up: First, we describe how the intake of police reports and their follow-up investigations are currently regulated. Second, we argue how a dialogue-based approach could be used to model this. Third, we define the framework. And finally, we apply the framework to cases of online trade fraud.

2 Process description of an e-fraud case

2.1 Current state of the report intake system

In the Netherlands, the reporting of complaints about online trade fraud is currently regulated as follows [12]: The applicant is presented with a multiple choice option field containing different complaint scenarios, some of which lead to intake forms while others lead to an explanation of why that specific scenario is a civil matter instead of a criminal one (see Table 1). The intake form is matched to the applicant's digital ID. They are asked to fill in their own personal info, as well as that of the counterparty, insofar as this is known to them (see Table 2). Afterwards, they are requested to give a description of the conflict in an open text field, as well as some additional information surrounding the transaction (see Table 3). After the form is filed, it is submitted to an online police database and the applicant may be contacted by phone or receive notifications through e-mail [7]. Their report is sent to their home address per mail, which they have to sign and send back to the police in order to confirm their complaint.

Scenario	**Valid?**
Nothing has been delivered 5 days after the promised delivery date.	*yes*
The delivered product is not as agreed.	*no*
The delivered product broke after a short while.	*no*
The delivered product turned out to be fake.	*yes*
The delivered product did not meet expectations.	*no*
No payment has been received after delivering the product.	*yes*

Table 1: Possible scenarios, and whether they make for valid trade fraud claims.

Personal data	**Applicant**	**Counterparty**
First name(s)	*required*	*optional*
Surname prefix	*optional*	*optional*
Surname	*required*	*optional*
Gender	*required*	*optional*
Date of birth	*required*	*optional*
Place of birth	*required*	*optional*
Street name	*required*	*optional*
House number	*required*	*optional*
House number suffix	*optional*	*optional*
Residence	*required*	*optional*
Zip code	*required*	*optional*
E-mail address	*required*	*required*
Phone number	*required*	*optional*
Cell phone number	*optional*	*optional*

Table 2: Required and optional data in the intake form, to be filled in by the applicant.

2.2 Dutch trade fraud laws

The official police guidelines for managing online fraud complaints distinguish between four types of scam resources [2]. These stratagems, originally expressed in legal jargon from the Dutch criminal code[5], will henceforth be referred to as:

1. Adopting a fake name (*het aannemen van een valse naam*)
2. Adopting a fake mark (*het aannemen van een valse hoedanigheid*)
3. Chicanery (*listige kunstgrepen*)
4. A web of lies (*een samenweefsel van verdichtsels*)

[5] Article 326, *Wetboek van Strafrecht*.

Information	Type of field	Required?
Does it concern a trade website (and which one)?	*multiple choice*	*yes*
Title of advertisement	*open text field*	*no*
Advertisement number	*open text field*	*no* [1]
Applicant's user name	*open text field*	*yes*
Counterparty's user name	*open text field*	*no*
Conflict description	*open text field*	*yes*
Applicant's bank account number	*open text field*	*no*
Transaction date	*insert date*	*yes*
Transaction time	*insert time*	*no*
Purchase money	*insert number*	*yes*
Transaction method	*multiple choice*	*yes*
Bank account number counterparty [2]	*open text field*	*yes*
Account holder counterparty [3]	*open text field*	*yes*
PayPal e-mail address counterparty [4]	*open text field*	*yes*
Permission to provide your e-mail address to counterparty	*multiple choice*	*yes*

Table 3: Conflict specification, to be filled in by the applicant.

If someone uses any of these four schemes to make anyone give away any of their possessions, provide any service, release any information, get into debt or remit someone else's debt, they can be convicted of fraud. The purport of the four resources is elaborated upon in an internal police document [7]. *Adopting a fake name* only applies to a person's proper name, not to the name in their e-mail address. One adopts a *false mark* if they use socially accepted behaviour in a deceitful way. Many scam methods can be considered *chicanery*, meaning that the victim is deceived through a ruse. *A web of lies* is a tight cohesion of falsehoods presented to the victim as facts. The victim is presented a misrepresentation of the situation to lure them into spending their money.

The case described in Figure 1 is an example of online trade fraud [7].

The suspect repeatedly used trade website Marktplaats to obtain event tickets without payment, while using the fake names 'John' and 'Thomas'. By sending an edited screenshot, he made it appear as if a transaction from his bank account to that of the disadvantaged party had taken place, under the guise of payment for the tickets. The disadvantaged parties subsequently sent their tickets to an e-mail address matching the fake name used by the suspect. The suspect used several different phone numbers to conceal his real identity and to create the impression that he would be reachable in case of problems. Since one of these phone numbers was registered in the police database, it was used to trace his real identity. After the police claimed access to his Marktplaats account, it turned out that he used multiple untraceable IP addresses and a single traceable one, that he was registered to Marktplaats with a single e-mail address containing his real name and that he used six different phone numbers. In total, six disadvantaged parties filed a complaint online, three of which completed their intake by signing the case file and sending it back to the police.

Figure 1: Anonymous summary of case file LMIO756

In this specific situation, the culprit's identity was discovered when one of his phone numbers matched one in the police database, and his fraud crimes were linked through his Marktplaats account. Other examples of information useful for tracing identity and linking crimes is bank account numbers, residence information and IP addresses.

Besides using a fake name in his transaction, the culprit also adopted a fake mark, namely that of a bona fide customer. His 'chicanery' was sending forged screenshots of bank transactions to his victims to convince them he had transferred the money.

[2] Required field if the concerned trade website is the popular Dutch trade platform Marktplaats.
[3] Only asked if IDEAL or money transfer was selected as the transaction method.
[4] Only asked if IDEAL or money transfer was selected as the transaction method.
[5] Only asked if PayPal was selected as the transaction method.

2.3 Case file construction

After the intake of a report, the follow-up e-fraud investigation is done by the LMIO. If they are able to match a number of submitted reports to the same culprit, then the next task is to figure out their identity. By contacting external sources, such as the trade website the suspect has used to allegedly scam their victims or the bank to which the bank account the suspect has used belongs, the police can obtain valuable information, such as personal details (name, address, residence), e-mail addresses or IP addresses. When enough relevant information has been gathered, LMIO constructs a case file that describes the alleged modus operandi of the suspect, along with the details on how the evidence has been obtained. This case description can then be used as grounds for the arrest and prosecution of the suspect. Currently, for the prosecutor's office to consider prosecution, a suspect should be linked to at least three cases of fraud.

It's possible that an investigation turns out that the alleged scammer and the owner of the used bank account(s) are in fact two or more different people. This means that the bank account holder could (knowingly or unknowingly) be an accomplice to a crime as they have laundered money for the scammer[6].

2.4 Different roles in an e-fraud case

In the process starting with the intake and ending at the case file, we distinguish between seven roles for all involved parties.

Police: [7]

Intake regulator. Establishes whether police reports presumably involve a case of fraud, and if not, whether the applicant has the option of starting a civil case. They furthermore ensure that the submitted information is complete.

Internal investigator. Searches through the police databases to check if any of the information in the reports matches that of previous cases.

External investigator. Inquires information about the case from specific third parties, depending on the information in the report.

Case file author. Responsible for combining all the acquired information into a single file.

Others involved:

Applicant. An alleged victim of fraud who has submitted a police report.

Counterparty. The person(s) or website accused of fraud by the applicant.

Third parties. Trade websites, banks, the Dutch population register ('Basisregistratie Personen'), social networks, internet providers and other organisations that may have relevant information about the conflict or its involved parties.

The different roles described here are assumed to be separate agents, such that they can engage in dialogues with one another. Police employees fulfilling one of the four roles will be modelled as one of three types of agents: An *intake agent* has the role of intake regulator, an *analysis agent* has the role of case file author and an *inquiring agent* can be an interal or an external investigator.

[6] Article 420bis, *Wetboek van Strafrecht.*

[7] Although we define these as seperate roles, it's possible that someone has more than one role or that the tasks of a single role are divided among multiple persons.

3 Inspirations from established dialogue types

The dialogue framework for intake and case filing we will construct in this thesis, takes inspiration from the more general dialogue types. Parsons, McBurney & Wooldridge [9] consider three basic types of formal inter-agent dialogue, based on categorization by Walton & Krabbe [18], namely *information-seeking dialogues*, *persuasion dialogues* and *inquiry dialogues*. Each of these types has their own legal game rules and goals. We will briefly discuss the specifics of the three games and in what way they serve as a basis for our dialogue model.

3.1 Information-seeking dialogues

An information-seeking dialogue presumes an asymmetric relation between agents. One agent attempts to extract an answer to one or multiple questions from the other [11]. The police agents are supposed to both gather information and make judgement calls based on said information, and this framework appears to support that purpose. Agent 1, the information seeker, initiates the conversation by asking 'p?', meaning 'is it true that p?' [9]. Agent 2 responds by asserting p, $\neg p$ or $\mathcal{U}$ (unable to answer). Agent 1 can accept its opponent's reply, or challenge it. They may not question the inability to answer. Acceptance terminates the dialogue, challenging requires Agent 2 to back up their claim with an argument. Once again, Agent 1 can accept or challenge. This procedure repeats itself until agent 1 concedes the claim or until Agent 2 retracts it. A successful information-seeking dialogue will end with one agent getting the other to accept p or $\neg p$ [10].

Characteristic of the information-seeking dialogue is the fact that the two roles remain the same over the course of the game. This is a property we want to implement in at least one section of the intake dialogue, which we will call the *specification exchange*. Here, one player (the intake agent) will attempt to extract answers to questions from its opponent (the human applicant) concerning the formalities of the conflict (any missing information from Tables 2 and 3). Until enough information is gathered and this component of the dialogue terminates, the intake agent retains the role of the interrogator and the applicant keeps the role of respondent. If the interrogator has any questions about specific circumstances of the conflict, it will ask the respondent about it, and depending on the reply, concede or press for more information. Once it receives an acceptable answer, a new round starts for the same procedure, and this is repeated until the information it requires is complete.

In some specific situations where information is missing from an intake, we find that the original protocol is inadequate for obtaining additional information. The specification exchange protocol should be an expansion of the one Parsons et al. [9] describe in the sense that not only challenges (why-questions) are a valid method of asking the other player to elaborate on their claims, but also *specification questions*. In the situation that the applicant has stated that they possess a specific piece of information, but have not stated exactly what this is, the intake agent should be able to ask for specification. For instance, if the applicant states that they know the phone number of the counterparty, the logical next step of the intake agent is to ask what this phone number is.

Information-seeking dialogue: example 1

P_1: "I know the counterparty's phone number."

P_2: "What is it?"

P_1: "06-3141593."

P_2: "I concede."

Classical information-seeking dialogue allows a player to respond to a question with asserting $\mathcal{U}$, meaning that they, for some reason, are unable the answer. This cannot be challenged and terminates the dialogue. We wish to redefine $\mathcal{U}$ in a way that it is not an incontrovertible statement. Uncertainty about knowledge is a prevailing reason for leaving out information in the intake form. We implement an adaptation that allows the challenging of ignorance.

Information-seeking dialogue: example 2

P_1: "Did you have contact over the phone with the counterparty?"

P_2: "I cannot answer that." ($\mathcal{U}$)

P_1: "Why not?"

P_2: "I'm not sure that it was the counterparty."

We should keep in mind that at the end of the dialogue the intake agent is not required to know every possible detail about the counterparty and the conflict, but it is required to know *whether the applicant knows it or not*. This is an important distinction.

3.2 Persuasion dialogues

In a persuasion dialogue, one agent tries to persuade the other to adopt a belief they do not currently hold [11]. The goal of the dialogue is to resolve a conflict of points of view [14]. One could argue that for every fraud claim, the applicant tries to persuade the police to adopt the belief that fraud has been committed. Generally, the game starts with Agent 1 asserting a statement p. Agent 2 will then either concede p, challenge p or claim $\neg p$. Acceptance will terminate the dialogue. The challenge will result in Agent 1 submitting an argument in favour of p, which initiates another round of the game starting with said argument. The statement $\neg p$ initiates a role-reversal where Agent 2 tries to persuade Agent 1 of its new claim. Prakken's liberal dialogue [13], a type of persuasion dialogue, differs from the aforementioned protocol in the sense that an agent is not only able to offer a counter-claim to its opponent's claims, but can also attack the inference of a conclusion based on its premises through an undercut. As will be discussed more thoroughly in the subsection on argumentation conflict, attacks result in arguments being justified, overruled or defensible.[8] If an argument is justified at the end of the dialogue, then we can gather that the conclusion follows from its premises. This property is well applicable to intake and case file dialogues, since it is a useful measurement in determining whether fraud can be presumed based on information known about the conflict.

In the discussion of the information-seeking dialogue, we discussed how the intake dialogue requires a relatively straightforward asymmetric interrogation in which the intake agent aims to establish completeness of information. We named this part of the dialogue the specification exchange. In addition to this, our dialogue also requires a *validation exchange* to make sure that the fraud claim is valid, i.e. that the conflict is one in which it appears that the counterparty has acted fraudulently towards the applicant.

The validation exchange is based mainly on the set-up of the persuasion dialogue. The first move of the game is an information dump by the applicant: they assert that they have a valid case for fraud, and put forward a number of claims in the conflict description field of the intake form that often do not form a coherent argument. If, based on these claims, the intake agent can construct a justified argument with the conclusion that the applicant may very well be a victim of online trade fraud, it will *accept* the validity of the fraud claim, and the validation exchange terminates. If not, it can either challenge or oppose the claim, giving the applicant the opportunity to clarify their reasoning. Liberal dialogues have the rule that a dialogue should begin with a claim or an argument [13]. We intend to adopt this property, but in most cases, there are several claims and arguments the intake agent can respond to based on the submitted report. It should run through them one by one, and each argument it challenges or can provide a counterargument to, initiates a sub-dialogue within the validation exchange. The agents are allowed to respond to claims and arguments made at an earlier points in time. If at some point in the conversation the intake agent is persuaded that the fraud claim is valid, this part of the dialogue is complete. This is the case if the statements by the applicant can form an effective argument for accepting the validity of the fraud claim based on the argumentation rules.

Although the option to ask questions to the conversational partner about certain topics, like in information-seeking dialogues, is generally not present in persuasion dialogues, we suggest adding this to give the intake agent the option to ask the applicant about missing details of the conflict.

See the example below for an illustration of challenging and concession in the context of the validation exchange.

[8] See section 4.2.

Persuasion dialogue: example

P_1: "The counterparty broke contact after the transaction."

P_2: "What do you mean by that?"

P_1: "We spoke over Whatsapp prior, but he blocked me afterwards."

P_2: "I concede."

3.3 Inquiry dialogues

A successful inquiry dialogue makes both agents reach a conclusion together that they would not have been able to reach on their own [11]. The inquiry protocol is relevant in the sense that intake agents and analysis agents should inquire whether a situation involves fraud. Agent 1 starts the dialogue by asserting an argument A. Agent 2 may concede or challenge the argument. When challenged, Agent 1 should give some support for its reasoning. If one of the arguments proves acceptable to both agents, the dialogue terminates. If not, the agents reverse roles. A general framework for inquiry dialogues has been presented by Black & Hunter [5], such that the agents are allowed to jointly construct arguments. They formally define that the outcome of a well-formed inquiry dialogue can be constructed from the union of both agents' beliefs. This conversational structure can be described as an attempt by two agents to unravel the truth together, which is also a goal for the agents in the dialogue system we aim to model.

The inquiry dialogue protocol is especially useful for the case file dialogue. For example, if someone from LMIO were to contact a bank for an information request about the identity of a specific bank account holder, they should be able to argue why they need this information, without giving away sensitive data about the case. The analysis agent has at this point received the strongest argument in favour of the fraud claim from the intake agent, containing everything known about the applicant, the conflict and the counterparty based on the intake. Since an inquiring agent should be able to provide an argument for an information request, it can inquire this from the analysis agent. See Figure 2 for an informal illustration of such an argument.

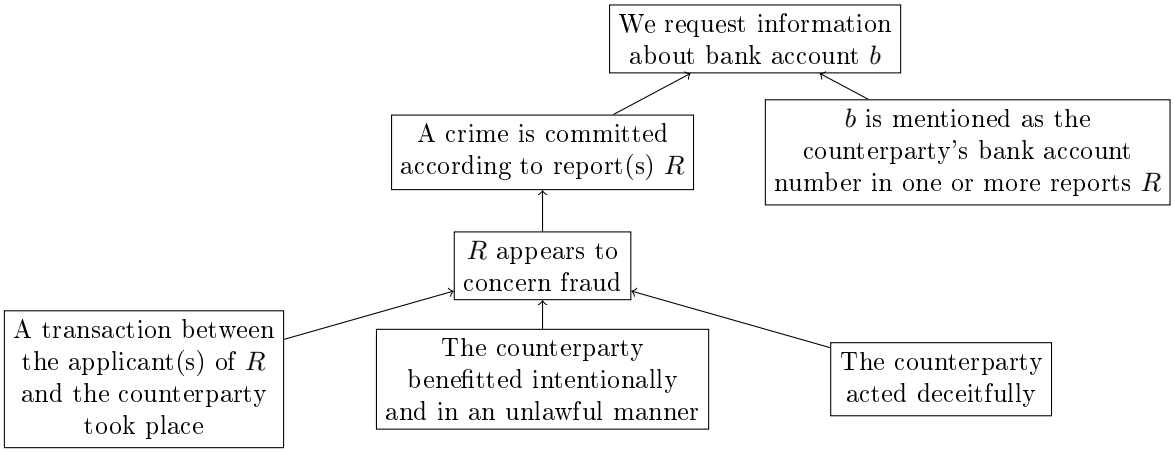

Figure 2: A simple, informal argument for the information request of a bank account.

3.4 Unique dialogue properties

Besides using somewhat modified properties of the predefined dialogue protocols to construct argumentations for fraud claims, we also need to implement a number of features not present in the common dialogue types.

In a certain sense, the shared goal of all agents in the dialogue is to construct a fitting narrative of the conflict. Bex & Prakken [4] have done something similar by combining story-telling and argumentation in a dialogue game, such that through the use of evidence, abductive inferences about incomplete scenarios are allowed. The final, justified argument in the case file should be translatable into a coherent story about the modus operandi of the suspect.

The case file dialogue requires a type of investigation request. If the moderator of the case file, which is the analysis agent, argues that there exists a 'loose thread' in the case, then it can send out either a general investigation request to all its inquiring agents in order to inquire more information about it from internal or external sources. The agent-specific request is a speech act directed towards a single inquiring agent, while he general request is something of the type "Who of you can investigate p?", where p is a lead on the identity of the counterparty, such as a phone number, an e-mail address or a bank account number. The inquiring agents, based on the type of information their source can provide, will answer either affirmatively or not.

Since the Dutch public prosecutor's office generally considers prosecuting someone after at least three police reports imply them as a suspect of fraud, the system needs to be able to match different police reports. This can be done by the analysis agent based on corresponding information in different reports. The case against the suspect is represented as a structured argument through combined information from the relevant police reports, which can be used as evidence in court. After receiving new reports, this argument can be modified to reflect the additional information. Besides making a case for fraud, the analysis agent could even implicate another party altogether for money laundering if it appears that the suspect used someone else's bank account to funnel the money.

The communication between intake and case filing should work both ways. Besides the intake agent presenting arguments to the analysis agent, the analysis agent should also be able to give feedback to the intake agent. For instance, the police sometimes use a whitelist for bona fide trade websites during manual intake, in order to reject complaints that most likely describe a default[9] by the website. Whether we want an automatic intake system to have the ability to give certain websites a free pass, is debatable, but the intake agent should have the option to at least temporarily exclude new cases based on knowledge provided through a dialogue with the analysis agent.

3.5 Combining dialogue types

What we've established through the informal indications of dialogue protocols in this subsection, is that the two dialogues in the model (the intake dialogue and the case file dialogue) should consist of separate exchanges, each with a different sub-goal and possibly an adapted set of legal locutions. Reed [15] models the embedding of dialogues through a locution that proposes a change of dialogue type, which conversational partners can accept or decline. This embedded structure fits the overarching conversational objective into a meta-level dialogue, which can be particularly useful for our model. This set-up allows the intake agent to switch between the specification exchange and the validation exchange, and allows the analysis agent to switch between conversation topics based on the findings of the different inquiring agents.

[9] 'Defaulting' is the failure to fulfil an obligation; something that falls under civil law, not criminal law, and therefore does not concern the police.

4 Preliminary definitions

Before discussing the notions specific to the intake and case file processes, we will formally define the relevant concepts of argumentation, conflict and dialogues.

4.1 Arguments and inferences

We use the $ASPIC^+$ framework as defined by Modgil & Prakken [8] to express our notions of argumentation. The following notation will be used throughout this thesis: Well-formed formulas (wff) are denoted by a Greek letter (such as ϕ, ψ, χ). If we want to specify a wff as a literal (a propositional variable or its negation), we use lowercase Roman letters (such as p, q, r). Since we use propositional logic to express formulas, 'wff' and 'proposition' will be used interchangeably.

Definition 4.1.1. A **defeasible inference rule** is a sequence of type $\psi_1, ..., \psi_n \Rightarrow \phi$.

Definition 4.1.2. A **strict inference rule** is a sequence of type $\psi_1, ..., \psi_n \rightarrow \phi$.

Strict inferences are deductive, meaning that the conclusion necessarily follows from the premises. The conclusion of a defeasible inference is something that follows typically, but not necessarily, from the premises.

We distinguish between two types of premises.

Definition 4.1.3. An **axiom premise** is a necessarily true assumptive proposition.

Definition 4.1.4. An **ordinary premise** is an uncertain assumptive proposition.

Through the application of inference rules given the premises, arguments are formed. An argument A is either a defeasible argument or a strict argument. A strict argument does not use any defeasible rules, and could therefore be simply a proposition. Argumentation theories dictate what arguments can be generated based on argumentation systems and knowledge bases.

Definition 4.1.5. An **argumentation system** AS is a triple $(\mathcal{L}, \mathcal{R}, n)$ where

- $\mathcal{L}$ is a logical language closed under negation,
- $\mathcal{R}$ is the union set of strict inference rules $\mathcal{R}_s$ and defeasible inference rules $\mathcal{R}_d$,
- n is a partial function such that $n : \mathcal{R}_d \longrightarrow \mathcal{L}$.

Definition 4.1.6. A **knowledge base** $\mathcal{K} \subseteq \mathcal{L}$ in an AS is the union set of known axiom premises $\mathcal{K}_n$ and known ordinary premises $\mathcal{K}_p$.

Definition 4.1.7. An **argumentation theory** AT is a tuple $(AS, \mathcal{K})$ where $\mathcal{K} \subseteq \mathcal{L}$ and $\mathcal{L}$ is contained in AS.

For arguments to be meaningful, they have to relate to a specific argumentation theory. The legality of premises, defeasible arguments and strict arguments is determined by the relevant knowledge base and argumentation system, according to the following seven functions:

1. $Prem(A)$ is the set of premises of an argument A,
2. $Conc(A)$ is the conclusion of A,
3. $Sub(A)$ is the set of sub-arguments of A,
4. $DefRules(A)$ is the set of defeasible rules applied in A,
5. $StrRules(A)$ is the set of strict rules applied in A,
6. $TopRule(A)$ is the top rule applied in A,
7. $Prop(A)$ is the set of propositions in A,
8. $VulProp(A)$ is the set of vulnerable propositions in A.[10]

[10] The vulnerable propositions are those that can be attacked according to definitions 4.2.1 and 4.2.2.

Definition 4.1.8. An argument A on the basis of an argumentation theory $AT = ((\mathcal{L}, \mathcal{R}, n), \mathcal{K})$ is one of three types:

1. A propositional argument $A = \phi$ if $\phi \in \mathcal{K}$ and

   $Prem(A) = \{\phi\}$,
   $Conc(A) = \phi$,
   $Sub(A) = \{\phi\}$,
   $DefRules(A) = \emptyset$,
   $StrRules(A) = \emptyset$,
   $TopRule(A) =$ undefined,
   $Prop(A) = \{\phi\}$,
   $VulProp(A) = \{\phi\}$ if ϕ is an ordinary premise,
   $VulProp(A) = \emptyset$ if ϕ is an axiom premise.

2. A defeasible argument $A = A_1, ..., A_n \Rightarrow \phi$ if $\{Conc(A_1), ..., Conc(A_n) \Rightarrow \phi\} \in \mathcal{R}_d$ and

   $Prem(A) = Prem(A_1) \cup ... \cup Prem(A_n)$,
   $Conc(A) = \phi$,
   $Sub(A) = Sub(A_1) \cup ... \cup Sub(A_n) \cup \{A\}$,
   $DefRules(A) = DefRules(A_1) \cup ... \cup DefRules(A_n) \cup \{Conc(A_1), ..., Conc(A_n) \Rightarrow \phi\}$,
   $StrRules(A) = StrRules(A_1) \cup ... \cup StrRules(A_n)$,
   $TopRule(A) = Conc(A_1), ..., Conc(A_n) \Rightarrow \phi$,
   $Prop(A) = Prop(A_1) \cup ... \cup Prop(A_n) \cup \{\phi\}$,
   $VulProp(A) = VulProp(A_1) \cup ... \cup VulProp(A_n) \cup \{\phi\}$.

3. A strict argument $A = A_1, ..., A_n \rightarrow \phi$ if $\{Conc(A_1), ..., Conc(A_n) \rightarrow \phi\} \in \mathcal{R}_s$ and

   $Prem(A) = Prem(A_1) \cup ... \cup Prem(A_n)$,
   $Conc(A) = \phi$,
   $Sub(A) = Sub(A_1) \cup ... \cup Sub(A_n) \cup \{A\}$,
   $DefRules(A) = DefRules(A_1) \cup ... \cup DefRules(A_n)$,
   $StrRules(A) = StrRules(A_1) \cup ... \cup StrRules(A_n) \cup \{Conc(A_1), ..., Conc(A_n) \rightarrow \phi\}$,
   $TopRule(A) = Conc(A_1), ..., Conc(A_n) \rightarrow \phi$,
   $Prop(A) = Prop(A_1) \cup ... \cup Prop(A_n) \cup \{\phi\}$,
   $VulProp(A) = VulProp(A_1) \cup ... \cup VulProp(A_n)$.

4.2 Conflict

When two arguments are contradictory, they are in conflict. In a conflict between two arguments, at least one of the arguments attacks the other. We distinguish between three types of attacks: the undermining attack (attacking an argument on one of its premises), the rebuttal (attacking an argument on its conclusion) and the undercutting attack (attacking the argument's inference of the conclusion based on the premises). Defeasible arguments may be attacked on their conclusions or inferences while strict arguments may not. Similarly, regular premises may be attacked while axiom premises may not.

Definition 4.2.1. Argument A **undermines** argument B if there exists a proposition ϕ such that $Conc(A) = \neg\phi$, $\phi \in Prem(B)$ and ϕ is not an axiom premise.

Definition 4.2.2. Argument A **rebuts** argument B if there exists a proposition ϕ such that $Conc(A) = \neg\phi$ and $TopRule(B') = B_1, ..., B_n \Rightarrow \phi$ for a $B' \in Sub(B)$.

Definition 4.2.3. Argument A **undercuts** argument B if there exists a defeasible rule d such that $Conc(A) = \neg n(d)$ and $d \in DefRules(B)$.

If one part of an argument is attacked, then its conclusion is also indirectly attacked. We call a successful attack a defeat. We assume in this thesis there are no self-defeating arguments. This success is dependent on the strength of an argument. Whether an argument is weaker or stronger than another argument is determined by the preference ordering over defeasible rules and regular premises.

Definition 4.2.4. The **preference ordering** $\preceq$ determines the preference of two defeasible rules or two regular premises x and y in relation to each other.

$y \prec x$: x is strictly preferred over y.

$y \preceq x$: x is either preferred over y or both are equally preferred.

$y \approx x$: x and y are equally preferred.

Definition 4.2.5. According to the **last-link** principle, an argument A is preferred over B if the top defeasible rules applied in A are preferred over the top defeasible rules applied in B.

Definition 4.2.6. According to the **weakest-link** principle, an argument A is preferred over B if the defeasible rule with the lowest preference applied in A is preferred over the defeasible rule with the lowest preference applied in B.

Definition 4.2.7. Argument A **defeats** argument B if A attacks B and if $B \preceq A$.

For both preference orderings it holds that if there are no defeasible rules applied to either argument, then A is preferred over B if A's premises are preferred over B's premises. Strict arguments have no preference relation.

Based on the defeat relationships between arguments, we assign a status to each argument. Throughout this thesis, we will assume *grounded semantics*, such that each argument can have only one status.

Definition 4.2.8. If an argument A is **justified** under grounded semantics, then it is either undefeated, or all arguments that defeat A are overruled.

Definition 4.2.9. If an argument A is **overruled** under grounded semantics, then there exists a justified argument B, such that B defeats A while A does not defeat B.

Definition 4.2.10. If an argument A is **defensible** under grounded semantics, then there exists a defensible argument B that defeats A. If A is defeated by any other arguments, then those are either overruled or defensible.

4.3 Dialogues

We apply the following definition of dialogue systems [14]:

Definition 4.3.1. A **formal dialogue system** $\mathcal{D}$ is a tuple ($\mathcal{L}_t$, $\mathcal{L}_c$, dp, $\mathcal{P}$, C, B, L, $\mathcal{K}$, $\mathcal{C}$, T, Pr, O), where:

- $\mathcal{L}_t$ is the topic language,
- $\mathcal{L}_c$ is the communication language,
- dp is the dialogue purpose,
- $\mathcal{P}$ is the participant set,
- C is the set commitment sets $C_i \in \mathcal{L}_t$ for all $P_i \in \mathcal{P}$,
- B is the set of belief bases $B_i \in \mathcal{L}_t$ for all $P_i \in \mathcal{P}$,
- L is the logic for $\mathcal{L}_t$,
- $\mathcal{K} \in \mathcal{L}_t$ is the context,
- $\mathcal{C}$ is the set of effect rules for $\mathcal{L}_c$,
- T is the turn-taking function,
- Pr is the protocol for $\mathcal{L}_c$,

- O is the set of outcome rules.

Definition 4.3.2. A **dialogue** d generated through a formal dialogue system $\mathcal{D}$, is a sequence in the communication language $\mathcal{L}_c \in \mathcal{D}$ according to the protocol $Pr \in \mathcal{D}$.

Earlier, we mentioned that the intention of this project is to construct a dialogue framework for crime reporting. This terminology is somewhat ambiguous, since $ASPIC^+$ can be defined as a framework for specifying (abstract argumentation) frameworks [8]. What we mean by a dialogue framework is this: an abstraction of multiple interconnected dialogue systems in which the communication language, effect rules, protocol and the like are predefined, but the topic language and the argumentation rules can be altered if necessary. The rationality behind this definition, is the fact that these properties allow us to set up dialogues for handling crime reports while not restricting us to a specific crime type and unchangeable (interpretations of) laws.

Since a dialogue is a sequence of moves, we can describe the location of a move based on the moves that are its predecessors and successors. The order of the sequence is chronological. If a move has one or more predecessors, then one of these predecessors is the target of the move. A target is an earlier move it reacts to. In a dialogue, we differentiate between several different categories of moves.

Definition 4.3.3. An **initiator** starts the dialogue, and therefore has no predecessors.

Definition 4.3.4. A **response** is a move that has a target by a different speaker.

Definition 4.3.5. A **follow-up** is is a move that has a target by the same speaker.

Definition 4.3.6. A **child locution** is the umbrella term for responses and follow-ups.

Definition 4.3.7. A **terminator** ends the dialogue, and therefore has no successors.

We allow for utterances to be responses or follow-ups to utterances that do not directly precede them in the chronological sequence. For example, take the dialogue tree in Figure 3. Assuming that the protocol is single-move (participants only make one move per turn), three different sequences can be the result of this tree: (1,2,3,4,5,6,7), (1,2,3,6,7,4,5) and (1,2,3,6,5,4,7). Note that the order of the sequence does not affect the nature of an utterance: 4 is always a response to 3, independent of whether or not it is a direct successor to 3.

Although initiators and child locutions are disjoint sets, a terminator must be either an initiator, a response, or a follow-up, depending on the protocol. If we assume utterance 5 in figure 3 to be a terminator for a single-move protocol, then (1,2,3,6,7,4,5) is the only possible dialogue.

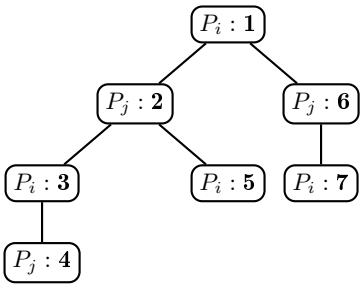

Figure 3: An example of a dialogue tree with 7 utterances. P_i and P_j denote the two participants. A child node is a successor to its parent node, and in this case is always a response. Child nodes of the same parent are ordered chronologically left-to-right, although this does not include their own offspring (for instance, 6 cannot precede 2, but it can precede 4).

4.4 Rule set

When reasoning about conflicts, we require a rule-based representation of the laws surrounding a conflict. Although the set-up of the dialogue systems we will define in sections 5 and 6 could be applied to conflicts of any nature, we focus specifically on e-fraud in this thesis. Therefore, any rule set $\mathcal{R}$ in an argumentation system AS,[11] needs to incorporate the guidelines given in section 2.2 while providing additional rules on how, based on the observations, one may conclude a fraud scheme as been applied. We define $\mathcal{R}_f$, represented in Table 4, as such.

We will briefly discuss the contents of $\mathcal{R}_f$. r_f1 strictly denotes that whenever someone has presumably committed fraud ($pFraud$), then they have presumably committed a crime ($pCrime$), since fraud is a crime. r_f2 states that $pFraud$ is established as having obtained goods with an intended fraudulent benefit[12]. r_f3 states that the intended fraudulent benefit is the breaking of a mutual agreement (r_f24 and r_f25) between the applicant and the counterparty by the latter, through an apparent fraud scheme while they make no effort to set it right (r_f26 to r_f39). From r_f4 and r_f5 it follows that if no goods have been obtained by the counterparty, then the applicant is not a victim of fraud. If the counterparty uses legitimate sale or purchase indicators to make it appear as if they are a bona fide salesman or a bona fide customer, then they employ a fraud scheme, see r_f6 and r_f7. For someone to be a deceitful salesman, they have to have offered a product and either not have sent it or have delivered a fake product instead, see r_f8 to r_f13, while giving the impression that their service is bona fide, see r_f15 to r_f20. For someone to be a deceitful customer, they have to have ordered something without paying, see r_f14, while giving a false indication of payment such as a (forged) transaction screenshot, see r_f21 to r_f23. r_f40 to r_f43 imply that someone can not be both the salesman and the customer for the same transaction.

[11] Definition 4.1.5.

[12] 'Wederrechtelijke bevoordeling', see Article 326, *Wetboek van Strafrecht*.

Rules				Preference
$r_f1:$	$pFraud$	$\rightarrow$	$pCrime$	
$r_f2:$	$IntendedFraudulentBenefit, ObtainingGoods$	$\Rightarrow$	$pFraud$	High
$r_f3:$	$FraudScheme, MutualAgreement,$ $BreakingContact$	$\Rightarrow$	$IntendedFraudulentBenefit$	High
$r_f4:$	$OrderedByA, PaymentByA$	$\Rightarrow$	$ObtainingGoods$	High
$r_f5:$	$OfferedByA, DeliveredByA$	$\Rightarrow$	$ObtainingGoods$	High
$r_f6:$	$DeceitfulSalesmanCP$	$\Rightarrow$	$FraudScheme$	High
$r_f7:$	$DeceitfulCustomerCP$	$\Rightarrow$	$FraudScheme$	High
$r_f8:$	$OfferedByCP, \neg SentByCP,$ $LegitimateSaleIndicator$	$\Rightarrow$	$DeceitfulSalesmanCP$	High
$r_f9:$	$OfferedByCP, SentByCP,$ $FakeProduct, LegitimateSaleIndicator$	$\Rightarrow$	$DeceitfulSalesmanCP$	High
$r_f10:$	$DeliveredByCP, \neg FakeProduct$	$\Rightarrow$	$\neg DeceitfulSalesmanCP$	High
$r_f11:$	$\neg DeliveredByCP, Waited5Days$	$\Rightarrow$	$\neg SentByCP$	High
$r_f12:$	$\neg DeliveredByCP$	$\Rightarrow$	$\neg SentByCP$	Low
$r_f13:$	$\neg Waited5Days$	$\Rightarrow$	$SentByCP$	Medium
$r_f14:$	$OrderedByCP, \neg PaymentByCP,$ $LegitimatePurchaseIndicator$	$\Rightarrow$	$DeceitfulCustomerCP$	High
$r_f15:$	$ProductPictured$	$\Rightarrow$	$LegitimateSaleIndicator$	High
$r_f16:$	$iDEALPayment$	$\Rightarrow$	$LegitimateSaleIndicator$	High
$r_f17:$	$TrackAndTrace$	$\Rightarrow$	$LegitimateSaleIndicator$	High
$r_f18:$	$SpecialOffer$	$\Rightarrow$	$LegitimateSaleIndicator$	High
$r_f19:$	$Webshop$	$\Rightarrow$	$LegitimateSaleIndicator$	High
$r_f20:$	$\neg DeliveredByCP$	$\Rightarrow$	$\neg LegitimateSaleIndicator$	Low
$r_f21:$	$TransactionScreenshot$	$\Rightarrow$	$LegitimatePurchaseIndicator$	High
$r_f22:$	$IntermediatePaymentPlatform$	$\Rightarrow$	$LegitimatePurchaseIndicator$	High
$r_f23:$	$\neg PaymentByCP$	$\Rightarrow$	$\neg LegitimatePurchaseIndicator$	Low
$r_f24:$	$OfferedByA, OrderedByCP$	$\Rightarrow$	$MutualAgreement$	High
$r_f25:$	$OfferedByCP, OrderedByA$	$\Rightarrow$	$MutualAgreement$	High
$r_f26:$	$ContactPrior, \neg ContactAfter$	$\Rightarrow$	$BreakingContact$	High
$r_f27:$	$\neg ContactPrior$	$\Rightarrow$	$\neg BreakingContact$	High
$r_f28:$	$ContactAfter$	$\Rightarrow$	$\neg BreakingContact$	High
$r_f29:$	$ContactOverTextOrWhatsapp$	$\Rightarrow$	$ContactPrior$	High
$r_f30:$	$ContactOverPhone$	$\Rightarrow$	$ContactPrior$	High
$r_f31:$	$ContactOverEmail$	$\Rightarrow$	$ContactPrior$	High
$r_f32:$	$ContactOverSocialMedia$	$\Rightarrow$	$ContactPrior$	High
$r_f33:$	$ContactThroughTradeWebsite$	$\Rightarrow$	$ContactPrior$	High
$r_f34:$	$\neg DeliveredByCP, \neg PaymentByCP$	$\Rightarrow$	$\neg ContactPrior$	Low
$r_f35:$	$BlockedByCP$	$\Rightarrow$	$\neg ContactAfter$	High
$r_f36:$	$WrongNumberCP$	$\Rightarrow$	$\neg ContactAfter$	High
$r_f37:$	$\neg AnswerPhoneCP$	$\Rightarrow$	$\neg ContactAfter$	High
$r_f38:$	$\neg RespondMessageCP$	$\Rightarrow$	$\neg ContactAfter$	High
$r_f39:$	$ContactPrior$	$\Rightarrow$	$ContactAfter$	Low
$r_f40:$	$OrderedByA$	$\Rightarrow$	$\neg OrderedByCP$	High
$r_f41:$	$OrderedByCP$	$\Rightarrow$	$\neg OrderedByA$	High
$r_f42:$	$OfferedByA$	$\Rightarrow$	$\neg OfferedByCP$	High
$r_f43:$	$OfferedByCP$	$\Rightarrow$	$\neg OfferedByA$	High

Table 4: Argumentation rule set $\mathcal{R}_f$ and its preferences, to be applied to establishing whether (presumably) the crime of online trade fraud has been committed. Loosely based on the rules of Bergers [1].

5 Intake dialogues

5.1 Dialogue set-up

The intake dialogue is a conversation between the applicant, who claims to be a victim of online trade fraud, and the intake agent, that has the purpose of ensuring the police report is correctly filed. Earlier, we described the desired properties of this dialogue type separately. Combined, the intake dialogue is set up such that it plays out as follows:

$$\text{Intake dialogue} = \begin{cases} \text{Initiate dialogue} \\ \text{Validation exchange (determine } pCrime) \\ \text{if } pCrime \begin{cases} \text{Specification exchange (determine } Complete) \\ \text{if } Complete\text{: accept } Intake \end{cases} \\ \text{Present argument} \\ \text{Terminate dialogue} \end{cases}$$

In the validation exchange, the intake agent intends to find out whether the conflict that the applicant describes presumably involves a crime (fraud in our examples). The conclusion is denoted by $pCrime$ if so, or $\neg pCrime$ if not. The way the intake agent must reach its conclusion is twofold: The first way is on the basis of claims the applicant made through the intake form prior to the dialogue. Claims that the intake agent deems acceptable are conceded. If a claim is unclear or ungrounded, the intake agent presses for more information by challenging it. If a claim appears to be wrong or contradictory to earlier statements, the intake agent can attack it through a counterclaim or counterargument. The applicant is free to present new claims at any point during the validation exchange. The second way through which the intake agent can build its case, is by asking the applicant directly what they think about certain matters. They can respond approvingly or dismissively, either through claims or arguments, or alternatively state that they are unable to answer ($\mathcal{U}$). Once again, the intake agent may press these claims if necessary. Should the intake agent conclude $\neg pCrime$, the dialogue terminates directly. If instead it concludes $pCrime$, the dialogue proceeds into the specification exchange.

The specification exchange is meant to fill in any gaps of information in the statement of the applicant. The current intake form already distinguishes between required and optional fields, but there are at least two types of situations imaginable in which it would be useful ask the applicant about certain missing information: First, some important fact about the conflict could be missing but not asked explicitly in the intake form. For example, the type of product or service over which the conflict has arisen is often missing in the submitted report. Second, the applicant could have omitted information that was in fact known to him. It is not so much required that all matters of the conflict are known to the applicant, but rather that it is *known if they are known*. Unique to the specification exchange is the fact that the intake agent can ask for a specification of the answer of the applicant. The correct response to this is by giving an instance of the subject in question. For example, if the intake agent inquires whether the applicant knows the bank account number of the counterparty, and they answer accordingly, the intake agent can ask for a specification, to which the applicant can respond with the actual bank account number. Once the intake agent has gathered all the data it requires, it can conclude $Complete$, meaning that the relevant information is complete.

If both $pCrime$ and $Complete$ hold, then the intake agent should conclude $Intake$, which denotes the acceptance of the police report intake. This terminates the dialogue.

5.2 Formal definition

Definition 5.2.1. An **intake dialogue system** for generating intake dialogues $\mathcal{D}^{id}$ is a tuple $(\mathcal{L}_t^{id}, \mathcal{L}_c^{id}, dp^{id}, \mathcal{P}^{id}, C^{id}, B^{id}, L^{id}, \mathcal{K}^{id}, \mathcal{C}^{id}, T^{id}, Pr^{id}, O^{id})$.

Definition 5.2.2. The **topic language** $\mathcal{L}_t^{id}$ is the set of propositions representing declarative sentences relevant to the crime. We use five types of propositions:

1. Literals,
2. $instance(\phi, \psi)$, meaning 'ψ is an instance of (category) ϕ',

3. $known(\phi)$, meaning 'ϕ is known',
4. $check(\phi)$, meaning 'we validated whether ϕ is known',
5. $\mathcal{U}(\phi)$ is a statement about ϕ and should be read as 'unable to comment on ϕ'; different rules hold for $\mathcal{U}$-based propositions than for regular propositions.

These specific three literals will be used throughout the protocol:

1. *pCrime*, meaning 'a crime has presumably been committed'.
2. *Complete*, meaning 'the details of the case are complete'.
3. *Intake*, meaning 'the police report intake is successful'.

Definition 5.2.3. The **communication language** $\mathcal{L}_c^{id}$ is a set of speech acts. The speech acts are listed below, the required subjects are either propositions from the topic language or arguments built around these propositions.

1. Begin (sub)-dialogue: $begin(\phi)$
2. Questions: $ask(\phi)$
3. Statements: $claim(\phi)$
4. Challenges: $challenge(\phi)$
5. Arguments: $argue(A)$
6. Asking for specification: $asktospecify(\phi)$
7. Concessions: $concede(A)$ or $concede(\phi)$
8. Retractions: $retract(A)$ or $retract(\phi)$
9. Finish (sub-)dialogue: $finish(\phi)$

We sometimes use the shorthand notation $s_{P_i}(t)$ for saying that participant P_i is the speaker of speech act s with topic t.

Definition 5.2.4. The **dialogue purpose** dp^{id} is to establish whether the conflict presumably concerns a crime, and if so to correctly file a police report. Based on the nature of the crime, the purpose can be specified further.

Definition 5.2.5. The **set of participants** $\mathcal{P}^{id}$ is $\{P_{int}, P_{app}\}$, with the roles of *intake agent* and *applicant*, respectively.

Definition 5.2.6. The **set of commitment sets** C^{id} contains the commitment sets for both participants: $C_{int} \subseteq \mathcal{L}_t^{id}$ and $C_{app} \subseteq \mathcal{L}_t^{id}$.

Definition 5.2.7. The **set of belief bases** B^{id} contains the belief bases of both participants: $B_{int} \subseteq \mathcal{L}_t^{id}$ and $B_{app} \subseteq \mathcal{L}_t^{id}$. The intake agent's beliefs are established prior to the dialogue[13], while the applicant's belief base is blind, meaning that an external observer cannot know what is contained in B_{app}.

Definition 5.2.8. The **logic** L^{id} is $ASPIC^+$, the associated inference rules are contained in the rule set $\mathcal{R}^{id}$.[14]

[13] See section 6.6.
[14] See section 5.3.

Definition 5.2.9. The **context** $\mathcal{K}^{id}$ consists of the axiom premises. No axiom premises are used, so $\mathcal{K}^{id} = \emptyset$.

Definition 5.2.10. The **set of effect rules** $\mathcal{C}^{id}$ specifies for each speech act $s_{P_i}(t) \in \mathcal{L}_c$ its effects on the commitments of a participant. C_i^m denotes the commitment set of participant P_i during move m. The following effect rules hold:

1. $m : claim_{P_i}(\phi) \longrightarrow C_i^m = C_i^{m-1} \cup \{\phi\}$
2. $m : argue_{P_i}(A) \longrightarrow C_i^m = C_i^{m-1} \cup Prop(A)$
3. $m : concede_{P_i}(A) \longrightarrow C_i^m = C_i^{m-1} \cup Prop(A)$
4. $m : retract_{P_i}(A) \longrightarrow C_i^m = C_i^{m-1} \setminus Prop(A)$

All other utterances have no effect on the commitment sets. Initially, the applicant is committed only to $pCrime$, while the intake agent's commitment set is empty.

Definition 5.2.11. The **turn-taking function** T^{id} returns for each dialogue the participant-to-move. The following holds:

1. The intake agent moves first, and preserves its turn until it initiates the validation exchange or the specification exchange.
2. During the validation exchange and the specification exchange, a participant can do only one move per turn, unless said move is a concession or a retraction. Either one is to be combined with another move unless the sub-dialogue has been terminated.
3. The information given by the applicant in the intake form is automatically embedded in the dialogue as a response to the initiation of one of both exchanges, but counts as a single turn.
4. The intake agent is the participant-to-move after the termination of a sub-dialogue or one of the two exchanges.
5. The intake agent preserves its turn after presenting a concluding argument in either exchange.

Definition 5.2.12. The **protocol** Pr^{id} specifies the legal moves at each stage of the dialogue. For all moves m and dialogues d it holds that $m \in Pr(d)$ if and only if all of the following conditions are satisfied:

Part 1: Meta-dialogue protocol

1. If m is played by participant P_x, then it is the turn of P_x.
2. A participant P_x may only play a speech act of type $claim(A)$ or $argue(A)$ if $Prop(A) \in B_x$. Meaning, they cannot state anything that they do not believe.
3. m is not a repetition of an earlier move, meaning that two child locutions to the same utterance cannot be equal. The sole exception to this condition are responses and follow-ups to $begin(pCrime)$ and $begin(Complete)$, which are allowed to have two or more identical child locutions. This exception gives agents the ability to 'restart' a conversation on a sub-topic that had already been completed unsatisfactorily.
4. m is not a child locution to a concession or retraction.
5. m is not a response to a move that has previously had a concession or a retraction as a response.
6. If m is the initiation of the dialogue, then m is $begin(Intake)$ and played by P_{int}.
7. If m is a child locution to $begin(Intake)$, then it is a follow-up played by P_{int} and of one of the following types:

- $begin(pCrime)$
- $begin(Complete)$
- $argue(A \rightarrow Intake)$ s.t. $Conc(A) = \neg pCrime$
- $argue(A, B \rightarrow Intake)$ s.t. $Conc(A) = pCrime$ and $Conc(B) = Complete$

Meaning, the intake agent can initiate the validation exchange, initiate the specification exchange, or make a concluding argument.

8. If m succeeds $begin(pCrime)$ and $finish(pCrime)$ has not (yet) been played, then m is part of the validation exchange. $begin(pCrime)$ and $finish(pCrime)$ are meta-locutions and not part of the exchange. See part 2 for the relevant protocol of the validation exchange.

9. If m succeeds $begin(Complete)$ and $finish(Complete)$ has not (yet) been played, then m is part of the specification exchange. $begin(Complete)$ and $finish(Complete)$ are meta-locutions and not part of the exchange. See part 3 for the relevant protocol of the specification exchange.

10. If $finish(pCrime)$ has been played, then m is not a child locution to a move in the validation exchange. Meaning, it is illegal to react to the validation exchange once it has been terminated.

11. If $finish(Complete)$ has been played, then m is not a child locution to a move in the specification exchange. Meaning, it is illegal to react to the specification exchange once it has been terminated.

12. If m is $finish(Intake)$, then it is played by P_{int} and a follow-up to $argue(A)$, where $Conc(A)$ is $Intake$. This terminates the dialogue.

Part 2: Validation exchange protocol

1. The validation exchange could be seen as a set of sub-dialogues about specific topics. Each of these mini-dialogues is initiated by a child locution to $begin(pCrime)$ and must be terminated before P_{int} presents its argument about $pCrime$.

2. m does not target anything in an already terminated sub-dialogue.

3. A sub-dialogue about a topic ϕ is initiated through one of two ways, namely either through inquiry by P_{int}:

   - $ask(\phi)$,

   or through assertion by P_{app}:

   - $claim(\phi)$,
   - $argue(A)$ s.t. $Conc(A) = \phi$.

4. A sub-dialogue about a topic ϕ is terminated through one of three ways, namely either through concession by P_{int}:

   - $concede(\phi)$,
   - $concede(\neg\phi)$,
   - $concede(\mathcal{U}(\phi))$,
   - $concede(A)$ s.t. $Conc(A) = \phi \vee \neg\phi \vee \mathcal{U}(\phi)$,

   or through retraction by P_{app}:

   - $retract(\phi)$,
   - $retract(\neg\phi)$,
   - $retract(\mathcal{U}(\phi))$,
   - $retract(A)$ s.t. $Conc(A) = \phi \vee \neg\phi \vee \mathcal{U}(\phi)$.

After the termination of the sub-dialogue, one may no longer respond to or follow up on claims made during the sub-dialogue.

5. If m is a follow-up by P_{int} to $begin(pCrime)$ that is not the initiation of a sub-dialogue, then it is of one of the following types:

   - $argue(A)$ s.t. $Conc(A) = pCrime$,
   - $argue(A)$ s.t. $Conc(A) = \neg pCrime$.

   The intake agent will give a concluding argument on whether or not a crime has presumably been committed.

6. If m is a child locution to a move of type $ask(\phi)$, then it is a response of one of the following types:

   - $claim(\phi)$,
   - $claim(\neg\phi)$,
   - $claim(\mathcal{U}(\phi))$,
   - $argue(A)$ s.t. $Conc(A) = \phi$,
   - $argue(A)$ s.t. $Conc(A) = \neg\phi$,
   - $argue(A)$ s.t. $Conc(A) = \mathcal{U}(\phi)$.

   The speaker can make a claim or argument on whether or not ϕ.

7. If m is a response to a move of type $claim(\phi)$, then it is of one of the following types:

   - $concede(\phi)$.
   - $challenge(\phi)$,
   - $claim(\neg\phi)$,
   - $argue(A)$ s.t $Conc(A) = \neg\phi$,

   A claim may be conceded, challenged, or countered.

8. If m is a child locution to a move of type $argue(A)$, then it is a response of one of the following types:

   - $concede(A)$
   - $challenge(\psi)$, where $\psi \in Prem(A)$,
   - $claim(\neg\phi)$, where $\phi \in VulProp(A)$,
   - $claim(\neg r)$, where $r \in DefRules(A)$,
   - $argue(B)$ s.t. $Conc(B) = \neg\phi$, where $\phi \in VulProp(A)$,
   - $argue(B)$, s.t. $Conc(B) = \neg r$, where $r \in DefRules(A)$.

   An argument may be conceded, challenged, or countered.

9. If m is a follow-up to a move of type $claim(\phi)$, then it is of the following type:

   - $retract(\phi)$.

   The retraction of a specific locution is always a child locution to the original statement.

10. If m is a child locution to a move of type $challenge(\phi)$, then it is a response of one of the following types:

    - $argue(A)$ s.t. $Conc(A) = \phi$.

    One can only respond to a challenge with an argument in favour of the challenged proposition. If they cannot make such an argument, then they must retract their original claim.

11. If m is a follow-up to a move of type $argue(A)$, then it is of one of the following types:

- $retract(A)$
- $retract(\phi)$, where $\phi \in VulProp(A)$

12. If m is a child locution to $claim(\mathcal{U}(\phi))$, then it is a response of one of the following types:

    - $challenge(\mathcal{U}(\phi))$,
    - $concede(\mathcal{U}(\phi))$.

    When $\mathcal{U}$ is claimed, this may be challenged or conceded.

13. If m is a child locution to a move of type $challenge(\mathcal{U}(\phi))$, then it is a response of one of the following types:

    - $argue(A)$ s.t. $Conc(A) = \mathcal{U}(\phi)$,
    - $\neg argue(A)$ s.t. $Conc(A) = \mathcal{U}(\phi)$,
    - $retract(\mathcal{U}(\phi))$.

    What makes $\mathcal{U}$ unique, is the fact that one may refuse to back up the $\mathcal{U}$ claim, instead of having to argue or retract.

14. m is not a child locution to a move of type $\neg argue(A)$, where $Conc(A) = \mathcal{U}(\phi)$. This statement is a figurative dead end, and P_{int} must concede the earlier claim of $\mathcal{U}(\phi)$.

15. If m is a follow-up to a move of type $claim(\mathcal{U}(\phi))$, then it is of the following type:

    - $retract(\mathcal{U}(\phi))$.

16. If m is $finish(pCrime)$, then it is played by P_{int} and a follow-up to $argue(A)$ where $Conc(A)$ is either $pCrime$ or $\neg pCrime$. Therefore, establishing whether presumably a crime has been committed exits the validation exchange.

Part 3: Specification exchange protocol

1. The specification exchange could be seen as a set of sub-dialogues about specific topics. Each of these mini-dialogues is initiated by a child locution to $begin(Complete)$ and must be terminated before P_{int} presents its argument about $Complete$.

2. m does not target anything in an already terminated sub-dialogue.

3. A sub-dialogue about a topic ϕ is initiated through one of two ways, namely either through inquiry by P_{int}:

    - $ask(known(\phi))$,

    or through assertion by P_{app}:

    - $claim(known(\phi))$,
    - $claim(\neg known(\phi))$,
    - $claim(instance(\phi, \psi))$.

4. A sub-dialogue about a topic ϕ is terminated through one of three ways, namely either through concession by P_{int}:

    - $concede(known(\phi))$,
    - $concede(\neg known(\phi))$,
    - $concede(\mathcal{U}(known(\phi)))$,
    - $concede(instance(\phi, \psi))$,
    - $concede(A)$, where $known(\phi) = Conc(A)$

    or through retraction by P_{app}:

    - $retract(known(\phi))$,

- $retract(\neg known(\phi))$,
- $retract(\mathcal{U}(known(\phi)))$,
- $retract(instance(\phi, \psi))$.

After the termination of the sub-dialogue, one may no longer respond to or follow up on claims made during the sub-dialogue.

5. If m is a follow-up by P_{int} to $begin(Complete)$ that is not the initiation of a sub-dialogue, then it is of the following type:

   - $argue(A)$ s.t. $Conc(A) = Complete$,

   The intake agent will give a concluding argument on why the intake data is complete.

6. If m is a response to $begin(Complete)$, then it is played by P_{app} and of one of the following types:

   - $claim(known(\phi))$,
   - $claim(instance(\phi, \psi))$,
   - $argue(instance(\phi, \psi) \rightarrow known(\phi))$.

   These are the claims and arguments the applicant makes concerning missing information about themselves, the counterparty, or the conflict, either when filling in the fields of the intake form or during the dialogue itself.

7. If m is a child locution to $ask(known(\phi))$, then it is a response of one of the following types:

   - $claim(known(\phi))$,
   - $claim(\neg known(\phi))$,
   - $argue(instance(\phi, \psi) \rightarrow known(\phi))$,
   - $claim(\mathcal{U}(known(\phi)))$.

   An applicant can claim that they do or do not have information about subject ϕ, or, more intuitively, directly state what they know about ϕ. If they state $\mathcal{U}(known(\phi))$ then they are unable or possibly unwilling to say anything about the subject.

8. If m is a response to $claim(known(\phi))$, then it is of one of the following types:

   - $concede(known(\phi))$,
   - $asktospecify(known(\phi))$.

   If the intake agent wants to know what the applicant knows specifically about ϕ, they can ask the applicant to specify what they know. If not, they can just concede.

9. If m is a response to $claim(\neg known(\phi))$, then it is of the following type:

   - $concede(\neg known(\phi))$.

   Absence of knowledge can only be conceded.

10. If m is a follow-up to $claim(\phi)$, then it is of the following type:

    - $retract(\phi)$.

    The only possible follow-up to a claim is its retraction.

11. If m is a child locution to $asktospecify(known(\phi))$, then it is a response of the following type:

    - $argue(instance(\phi, \psi) \rightarrow known(\phi))$.

    In other words, they can specify that they know ψ about category ϕ.

12. If m is a child locution to $argue(instance(\phi, \psi) \rightarrow known(\phi))$ or $claim(instance(\phi, \psi))$ by P_{app}, then it is a response by P_{int} of one of the following types:

    - $concede(instance(\phi, \psi) \rightarrow known(\phi))$ or $concede(instance(\phi, \psi))$,
    - $argue(instance(\phi, \chi) \Rightarrow \neg instance(\phi, \psi))$, where $instance(\phi, \chi) \in C^{app}$,
    - $argue(instance(\mu, \psi) \Rightarrow \neg instance(\phi, \psi))$, where $instance(\mu, \psi) \in C^{app}$.

    The intake agent shall in most cases simply concede the specification. However, it may object. A possible counterargument could be that the applicant has at an earlier point given either a different specification to the same category ϕ (for instance, the counterparty's bank account number is stated twice, one of these contains a typo), or the same specification ψ to a different category (for instance, the noted bank account numbers of the applicant and the counterparty are the same).

13. If m is a follow-up to $argue(A)$, it is of type:

    - $retract(A)$.

14. If m is a child locution to $argue(instance(\phi, \chi) \Rightarrow \neg instance(\phi, \psi))$, then it is a response of one of the following types:

    - $retract(instance(\phi, \chi))$,
    - $argue(instance(\phi, \psi), instance(\phi, \chi) \Rightarrow \neg r)$, where $r = \{instance(\phi, \psi) \Rightarrow \neg instance(\phi, \chi)\}$.

    In the situation that the intake agent has pointed out that two different specifications have been given, the applicant can either retract the old specification or argue that both instances are true (for instance, if the counterparty used two different names during the contact).

15. If m is a child locution to $argue(instance(\mu, \psi) \Rightarrow \neg instance(\phi, \psi))$, then it is a response of one of the following types:

    - $retract(instance(\mu, \psi))$,
    - $argue(instance(\phi, \psi), instance(\mu, \psi) \Rightarrow \neg r)$, where $r = \{instance(\phi, \psi) \Rightarrow \neg instance(\mu, \psi)\}$.

    These options are similar to the ones described in the previous step. Instead of the instance itself, the category of specification differs.

16. If m is a child locution to $argue(A)$, where $Conc(A)$ is of type $\neg r$, then it is a response of the following type $concede(A)$. Meaning, the only option of the intake agent is to accept that both specifications hold.

17. If m is $finish(Complete)$, then it is played by P_{int} and a follow-up to $argue(A)$ where $Conc(A) = Complete$. This exits the specification exchange.

Definition 5.2.13. The single **outcome rule** $o_1 \in O^{int}$ states that the outcome of the dialogue is the concluding argument A_{Intake} (see section 5.5).

5.3 Argumentation rules

Any argument brought up during the intake dialogue must be constructed in accordance to the rules in the argumentation system[15]. We have previously defined $\mathcal{R}_f$, the rule set for reasoning about conflict situations concerning e-fraud, in Table 4. We require an additional rule set $\mathcal{R}_i$ specifically for the intake dialogue. Table 5 contains the rules of $\mathcal{R}_i$ which, in combination with $\mathcal{R}_i$ or an alternative conflict-related set, are necessary for having a functional intake dialogue.
The majority of rules in $\mathcal{R}_i$ have already been mentioned in the protocol subsection[16], with the exception of r_i3 and r_i4. The former states that lack of knowledge about a topic is a valid reason

[15] Definition 4.1.5.
[16] Definition 5.2.12, part 1, step 7 and 12; part 3, step 6, 7, 11, 12, 14 and 15.

Rules				Preference
$r_i1:$	$pCrime, Complete$	$\Rightarrow$	$Intake$	High
$r_i2:$	$\neg pCrime$	$\Rightarrow$	$Intake$	High
$r_i3:$	$\neg known(\phi)$	$\Rightarrow$	$\mathcal{U}(\phi)$	High
$r_i4:$	$\phi, \neg\phi$	$\Rightarrow$	$\mathcal{U}(\phi)$	High
$r_i5:$	$known(\phi)$	$\rightarrow$	$check(\phi)$	
$r_i6:$	$\neg known(\phi)$	$\rightarrow$	$check(\phi)$	
$r_i7:$	$instance(\phi, \psi)$	$\rightarrow$	$known(\phi)$	
$r_i8:$	$instance(\phi, \chi)$	$\Rightarrow$	$\neg instance(\phi, \psi)$	Low
$r_i9:$	$instance(\mu, \psi)$	$\Rightarrow$	$\neg instance(\phi, \psi)$	Low
$r_i10:$	$instance(\phi, \chi), instance(\phi, \psi)$	$\Rightarrow$	$\neg r_i8$ (for same ϕ, ψ, χ)	High
$r_i11:$	$instance(\mu, \psi), instance(\phi, \psi)$	$\Rightarrow$	$\neg r_i9$ (for same ϕ, ψ, μ)	High

Table 5: Rule set $\mathcal{R}_i$.

for refusing to talk about said topic. The latter states that if one has reason to believe something as well as its contrary, one may choose to remain silent about it. The attentive reader may have noticed that neither $\mathcal{R}_f$ nor $\mathcal{R}_i$ contain rules with $Complete$ as their consequent. To choose these rules, we need to know which information should be checked in order to achieve completeness (i.e. finishing the specification exchange). $\mathcal{R}_c$ (Table 6) is an example of a relevant singleton rule set.

Rules			
r_c1	$check(PhoneNumberA), check(NameCP),$ $check(UsernameCP), check(TradeWebsite),$ $check(TransActionDate), check(PurchaseMoney),$ $check(PaymentMethod), check(AccountHolderCP),$ $check(TransactionTime), check(BankAccountA),$ $check(ProductType)$	$\rightarrow$	$Complete$

Table 6: Rule set $\mathcal{R}_c$.

Definition 5.3.1. Given rule set $\mathcal{R}^{id} = \{\mathcal{R}_i \cup \mathcal{R}_f \cup \mathcal{R}_c\} = \{\mathcal{R}_s^{id} \cup \mathcal{R}_d^{id}\}$[17], topic language $\mathcal{L}_t^{id}$ and partial function $n^{id} : \mathcal{R}_d^{id} \longrightarrow \mathcal{L}^{id}$, we have an argumentation system AS^{id} for intake dialogues.

5.4 Strategy

According to the intake protocol[18], the dialogue terminates once P_{int} can give an argument for $Intake$ through application of r_i1 or r_i2. To be able to do that, either $\{\neg pCrime\}$ or $\{pCrime, Complete\}$ must be part of its belief base B_{int}[19]. Under which conditions either one is accepted, depends on the strategy of the intake agent. Specifically, when to concede and when to challenge. For this thesis, we assume an exhaustive strategy: The intake agent does not concede an argument unless all its premises are *observables*. Under observables, we understand all propositions that need not be substantiated but can be stated as an observation (true or false). Ω^{id} is the set of observables in $\mathcal{L}_t^{id}$. It contains the items summed up in Figure 4.
Keep in mind that '$asktospecify()$' is not the same as a challenge, and may still be moved as a response to observables of type $known(\phi)$. In an exhaustive strategy, the intake agent should always ask for specification once they learn that the applicant knows something about a certain topic category.

The intake agent should furthermore be careful in drawing their conclusions about $pCrime$. If they move '$argue(A)$' where $Conc(A) = \neg pCrime$, then this terminates the validation exchange[20]. This move should only be played if no observables exist in Ω^{id} that, if their status would be known,

[17] $\mathcal{R}_s^{id}$ and $\mathcal{R}_d^{id}$ denote the strict and defeasible rule subsets of $\mathcal{R}^{id}$, respectively (see Definition 4.1.5).
[18] Definition 5.2.12, part 1, step 7 and 12.
[19] Definition 5.2.12, part 1, step 2.
[20] Definition 5.2.12, part 2, step 5 and 16.

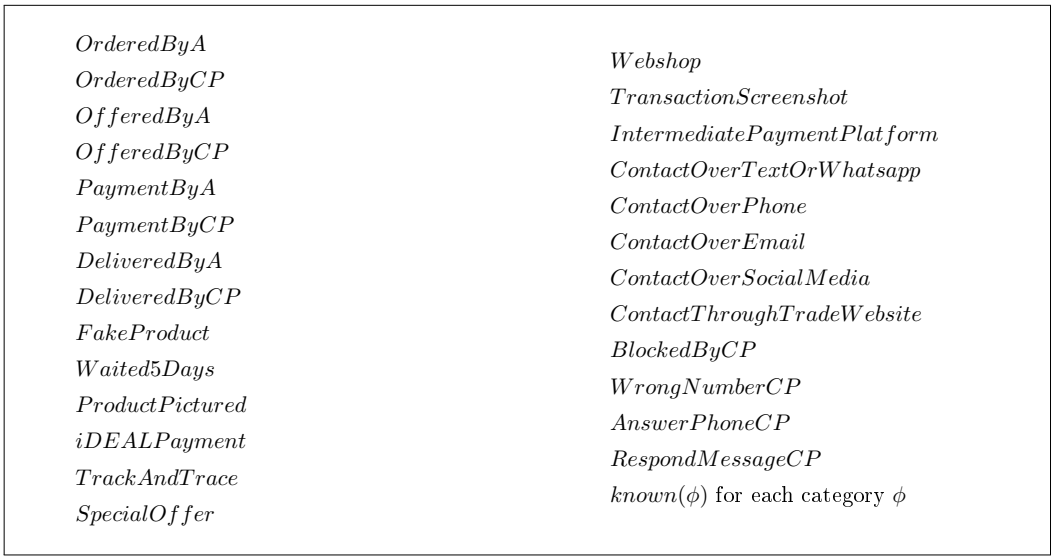

Figure 4: Observable set Ω^{id}.

could defeat A. If such a observable does exist, then the intake agent should ask about it before stating their argument.

Finally, the intake agent should not put forward any claim or argument that is not justified according to the weakest-link principle[21]. This prevents the drawing of conclusions that are not supported by the existing evidence.

5.5 Presenting a concluding argument

If the strategy of 5.4 is indeed applied, the concluding argument A_{Intake}, to be presented by the intake agent right before the termination of the dialogue, must:

1. have only observables and specifications of observables as their premises (if the specifications are known, they must be included),
2. be justified according to B_{int} under the weakest-link principle,
3. have $Intake$ as its conclusion.

5.6 Example

To demonstrate the workings of the intake dialogue system, we take the following set to represent the contents of a hypothetical police report:

$\{MutualAgreement, PaymentByA, ContactOverEmail, \neg ContactAfter,$
$instance(FirstNameCP, \text{'Johannes'}), instance(SurnamePrefixCP, \text{'de'}),$
$instance(SurnameCP, \text{'Silentio'}), instance(EmailAddressCP, \text{'fear@trembling.nl'}),$
$instance(BankAccountCP, \text{'NL99BANK01123581321'}),$
$instance(BankAccountCP, \text{'NL99BANK01123581325'}),$
$instance(TradeWebsite, \text{'Marktplaats'}), instance(PurchaseMoney, \text{'€270,-'})\}$

See Figures 5, 6, 7, 8 and 9 for a visualisation of an intake dialogue that be generated through $\mathcal{D}^{id}$ based on this report[22]. In this example, we simplify the established rules for the specification dialogue and replace $\mathcal{R}_c$ with $\mathcal{R}_{c'}$ (see Table 7). The applied rule set is $\mathcal{R}^{id'} = \{\mathcal{R}_f \cup \mathcal{R}_i \cup \mathcal{R}_{c'}\}$[23]. Figures 10 and 11 are visualisations of the arguments for $pCrime$ and $Complete$. The fully written out concluding argument A_{Intake} is enclosed in Appendix A.

[21] Definitions 4.2.6 and 4.2.8.
[22] Definition 5.2.1.
[23] Tables 4, 5 and 7, respectively.

	Rules		
$r_{c'}1$	$InfoCPComplete, InfoConflictComplete$	$\rightarrow$	$Complete$
$r_{c'}2$	$check(FirstNameCP), check(SurnamePrefixCP),$ $check(SurnameCP), check(EmailAddressCP),$ $check(PhoneNumberCP)$	$\rightarrow$	$InfoCPComplete$
$r_{c'}3$	$check(BankAccountCP), check(TradeWebsite),$ $check(PurchaseMoney)check(ProductType)$	$\rightarrow$	$InfoConflictComplete$

Table 7: Rule set $\mathcal{R}_{c'}$.

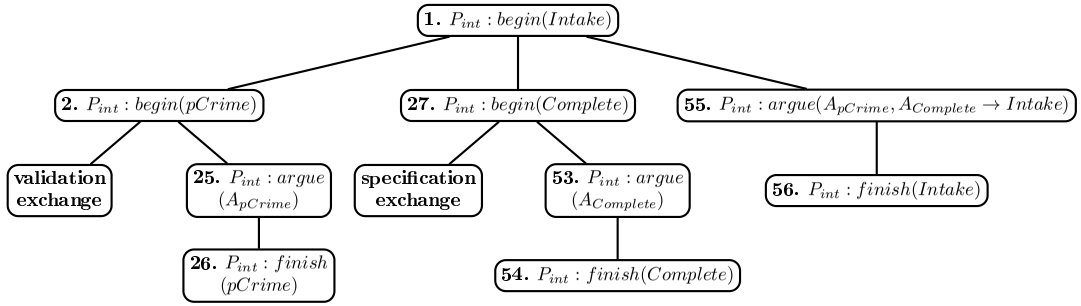

Figure 5: Example of an intake dialogue, part 1.

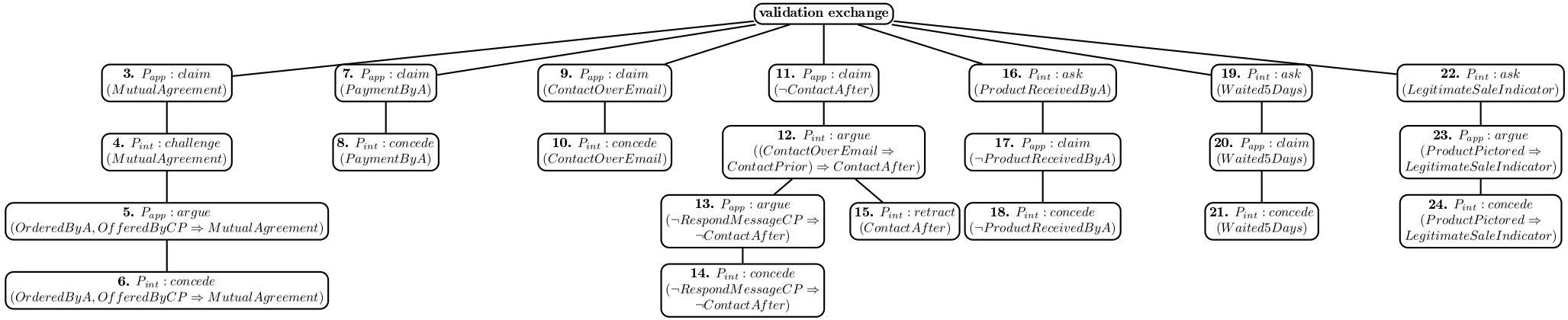

Figure 6: Example of an intake dialogue, part 2

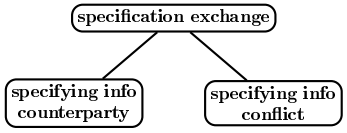

Figure 7: Example of an intake dialogue, part 3. Split up for readability.

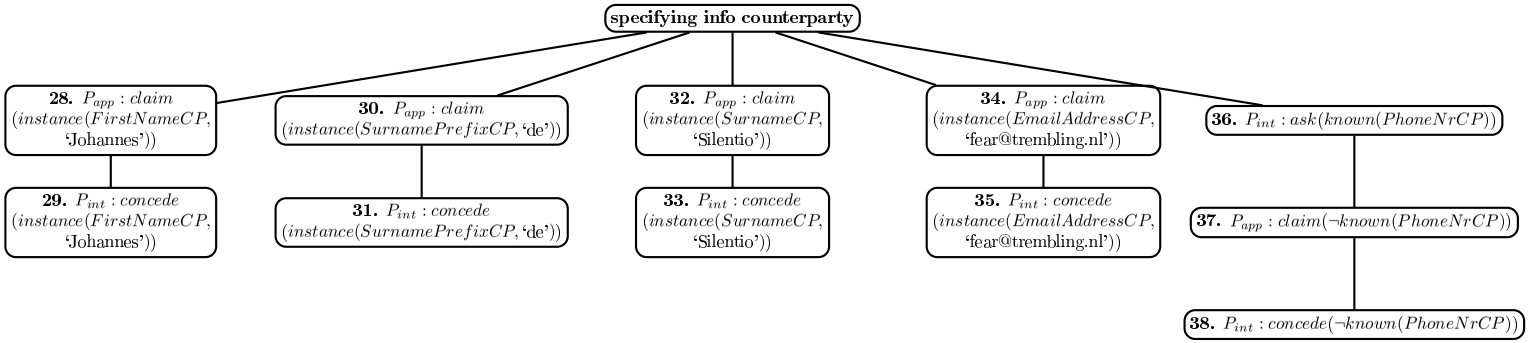

Figure 8: Example of an intake dialogue, part 4

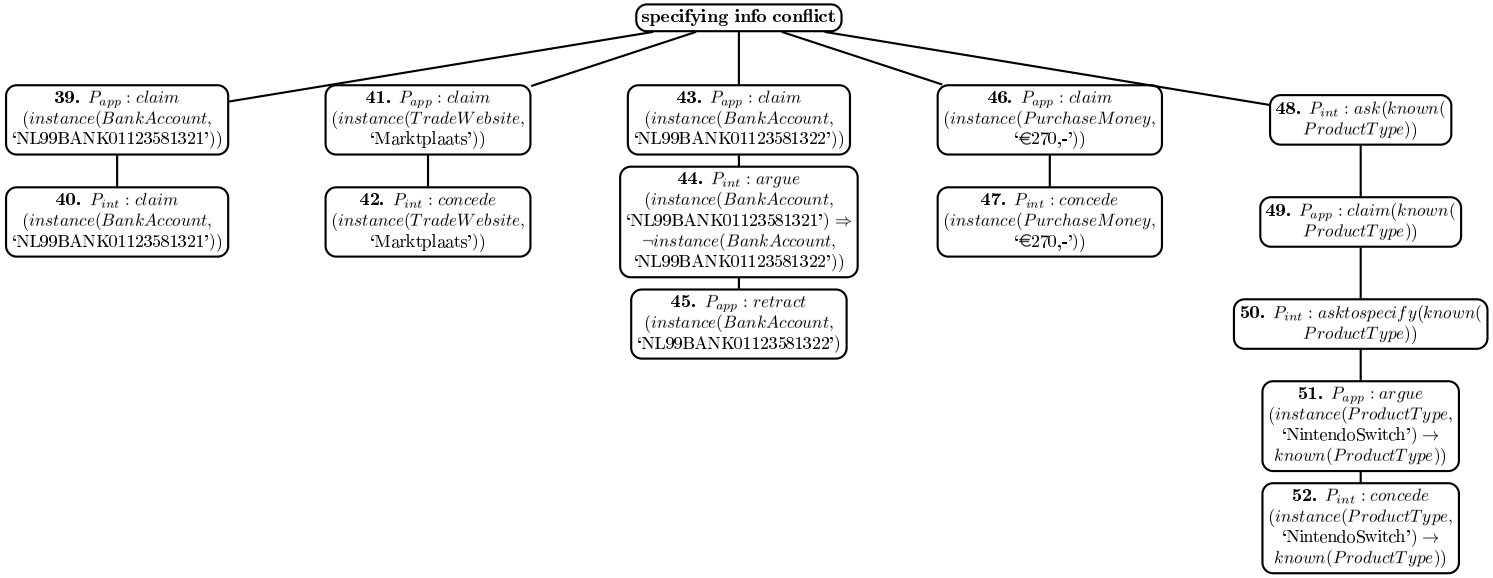

Figure 9: Example of an intake dialogue, part 5

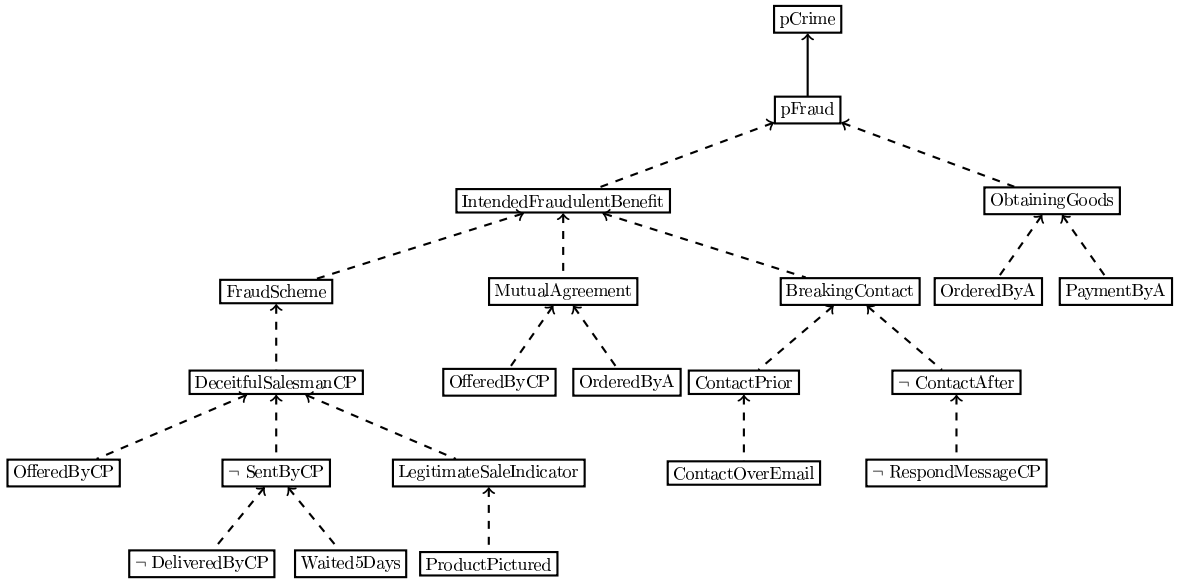

Figure 10: The argument A_{pCrime} referred to in Figure 5. Solid lines represent strict inferences, dashed lines represent defeasible inferences.

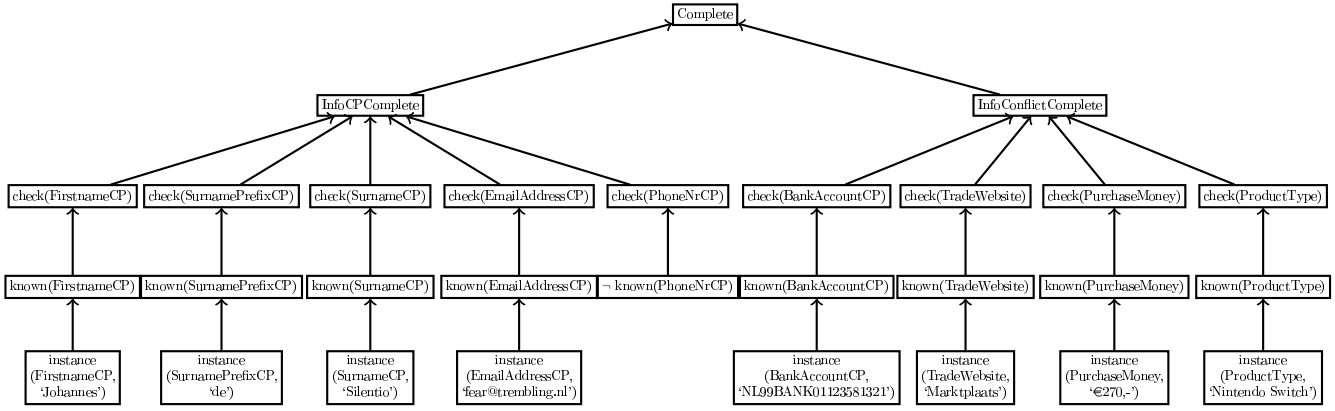

Figure 11: The argument $A_{Complete}$ referred to in Figure 5

5.7 Matching different reports

As mentioned earlier, a suspect should currently be linked to at least three cases of fraud before prosecution is considered. For that reason, a police investigation will only be performed after three or more reports appear to point to the same suspect. As this number is dependent on current regulation, which in turn assumes that all analysis (and prosecution) is done manually, an investigation could potentially be initiated based on any number of reports. Prior to the investigation, reports should be *matched* based on known information about the suspect. Keep in mind that all knowledge is based on the reports itself and that someone can easily assume multiple online identities, making it often difficult to accurately match conflicts based on a single suspect. Investigations could bring new information to light which connects instances of presumed fraud previously deemed unrelated. We distinguish between these two types of report matching: *pre-investigation* (between the intake process and the case file process) and *post-investigation* (during the case file process).

Since we want agents to be able to reason about the arguments in police reports, we define the following function q:

Definition 5.7.1. Given that $\mathcal{A}^{id}$ is the set of concluding arguments and their subarguments generated by $\mathcal{D}^{id}$: q is a partial function such that $q : \mathcal{A}^{id} \longrightarrow \mathcal{L}_t^{id}$.

Through q, concluding arguments made by the intake agent (see appendix A for an example of such an argument) will be given a label in the topic language. This labelling facilitates the matching of police reports, since arguments can be referred to as propositions. We will henceforth use $R_1, ..., R_n$ to denote (concluding arguments of) police reports. The reason q is a partial function is the fact that we only want the conflicts that presumably concern a crime to be considered by the case file agent. This means that all reports that contain *Intake* but not *pCrime* (as well as their sub-arguments) should not be passed along.

Pre-investigation report matching compares corresponding properties established in different reports and checks for similarities. We add the propositional types *match*() and *observed*() to our topic language. $match(R_1, R_2)$ means 'R_1 and R_2 are matched', and $match(A, R)$ means 'A can be observed in R'. The following holds:

Definition 5.7.2. If $A \in Sub(R)$ then we may conclude $observed(A, R)$. If $\phi \in Prop(R)$ then we may conclude $observed(\phi, R)$.

An example of a rule set for matching $\mathcal{R}_m$ that could be build around these two operators:

Rules				**Preference**
r_m1	$observed(instance(\text{BankAccountCP}, \phi), R_1)$, $observed(instance(\text{BankAccountCP}, \phi), R_2)$	$\Rightarrow$	$match(R_1, R_2)$	High
r_m2	$observed(instance(\text{TradeWebsite}, \phi), R_1)$, $observed(instance(\text{UserNameCP}, \psi), R_1)$, $observed(instance(\text{TradeWebsite}, \phi), R_2)$, $observed(instance(\text{UserNameCP}, \psi), R_2)$	$\Rightarrow$	$match(R_1, R_2)$	High
r_m3	$match(R_1, R_2)$	$\rightarrow$	$match(R_2, R_1)$	
r_m4	$match(R_1, R_2), match(R_2, R_3)$	$\rightarrow$	$match(R_1, R_3)$	

Table 8: Rule set $\mathcal{R}_m$.

The first two rules give potential grounds for matching. r_m1 states that two reports can be matched based on a single observation, namely the bank account. For r_m2, two observations are required, namely the trade website and the user name. These rules are obviously not exhaustible. r_m3 and r_m4 state that matching is a symmetric and transitive relation.

All reports submitted to the analysis agent by the intake agent are matched pre-investigation. The dialogue between the intake agent and the analysis agent has the simple structure of Figure 12, where the set of reports $\{R_1, ..., R_n\} \in \mathcal{A}^{id}$ is submitted, given that $n \geq 2$, along with the matching of said reports. P_{int} and P_{ana} denote the intake agent and the analysis agent, respectively.

1. $P_{int} : begin()$

2. $P_{int} : argue(R_1), ..., argue(R_n), claim(match(R_1, R_2)), ..., claim(match(R_{n-1}, R_n))$

3. $P_{ana} : concede(R_1), ..., concede(R_n), concede(match(R_1, R_2)), ..., concede(match(R_{n-1}, R_n))$

4. $P_{int} : finish()$

Figure 12: The structure of the dialogue between the intake agent and the analysis agent.

6 Case file dialogues

6.1 Dialogue set-up

The case file dialogue is a conversation between at least three participants: the analysis agent, a number of inquiring agents and equally as many external contacts. The analysis agent is the moderator, as it decides when the dialogue starts and when it ends. The case file dialogue is set up such that it plays out as follows:

$$\text{Case file dialogue} = \begin{cases} \text{Initiate dialogue} \\ \text{Investigation (iterated)} \begin{cases} \text{Information request} \\ \text{Information retrieval} \\ \text{Information report} \end{cases} \\ \text{Present argument} \\ \text{Terminate dialogue} \end{cases}$$

The analysis agent sends out one or more *investigation requests* to all of the inquiring agents to investigate a certain matter, such as the suspect's phone number or a bank account number, which each can then affirm or deny. Affirmation means that their external contact may have more information on the matter, denial means that they do not. If their external contact requires a substantiation on why they should give up this information, they can subsequently challenge the analysis agent to provide an argument in favour of this. This argument should be a description on why the person connected to the subject investigated is suspected to be in violation of the law, without giving up confidential information. This might result in a back and forth-discussion between the analysis agent and the inquiring agent, before the latter party concedes and starts talking to their external contact.

The ensuing sub-dialogue mirrors the previous exchange in the sense that the external contact must first affirm or deny the inquiry request, and after they affirm can ask for an argument in favour of providing the information. Once the inquiry agent has provided an argument the external contact deems sufficient, it will report anything it can share about the subject. For example, in the case that the external contact works for a bank, it will give up the personal information about the account holder. The inquiring agent will then report this back to the analysis agent.

If the analysis agent has gathered all it needs to know in order to construct a case file for the public prosecutor's office, it states its argument and terminates the dialogue.

6.2 Formal definition

Definition 6.2.1. A **case file dialogue system** for generating case file dialogues $\mathcal{D}^{cd}$ is a tuple $(\mathcal{L}_t^{cd}, \mathcal{L}_c^{cd}, dp^{cd}, \mathcal{P}^{cd}, C^{cd}, B^{cd}, L^{cd}, \mathcal{K}^{cd}, \mathcal{C}^{cd}, T^{cd}, Pr^{cd}, O^{cd})$.

Definition 6.2.2. The **topic language** $\mathcal{L}_t^{cd}$ is a set of propositions representing declarative sentences. We use three types of propositions:

1. Literals,
2. $instance(\phi, \psi)$, meaning 'ψ is an instance of (category) ϕ',
3. $relevant(\phi)$, meaning 'knowing about ϕ is relevant',
4. $information(\phi)$, which means that information about ϕ has been provided; for instance, if ϕ is a bank account number NL99BANK01123581321, then possible premises for an argument concluding with $information$(NL99BANK01123581321) are the personal details of the account holder,
5. $observed(\phi, R)$ or $observed(A, R)$, meaning 'ϕ or A is contained in report R',[24]
6. $matched(R_1, R_2)$, meaning 'reports R_1 and R_2 are matched'.

The specific literal $Casefile$ will be used throughout the protocol, which is the conclusion of the argument representing the case description.

[24] A and R, although they are arguments, are part of the topic language through function q, see section 5.7. The same holds for R_1 and R_2 in point 6.

Definition 6.2.3. The **communication language** $\mathcal{L}_c^{cd}$ is a set of speech acts. The speech acts are listed below, the required subjects are either propositions from the topic language or arguments built around these propositions.

1. Begin dialogue: $begin(\phi)$
2. Arguments: $argue(A)$
3. Challenges: $challenge(\phi)$
4. Group requests: $request(\phi)$
5. Affirmations: $affirm(\phi)$
6. Denials: $deny(\phi)$
7. Inquiries: $inquire(\phi)$
8. Reports: $report(A)$ where $Conc(A)$ is of type $information(\phi)$
9. Concessions: $concede(A)$
10. Retractions: $retract(A)$
11. Finish dialogue: $finish(\phi)$

We use the notation $s_{P_i 2 P_j}(t)$ for a shorthand notation of saying that speech act s with topic t is spoken by participant P_i towards participant(s) P_j.

Definition 6.2.4. The **dialogue purpose** dp^{cd} is to create a case file.

Definition 6.2.5. The **set of participants** $\mathcal{P}^{cd}$ is $\{P_{ana}, P_{inq_1}, ..., P_{inq_m}, P_{ext_1}, ..., P_{ext_m}\}$, with the roles of *analysis agent* for P_{ana}, of *inquiring agent* for $P_{inq_1}, ..., P_{inq_m}$ and of *external contact* for $P_{ext_1}, ..., P_{ext_m}$.

Definition 6.2.6. The **set of commitment sets** C^{cd} consists of the commitment sets $C_i \subseteq \mathcal{L}_t^{cd}$ for all participants P_i. Initially, all commitment sets are empty.

Definition 6.2.7. The **set of belief bases** B^{cd} consists of the belief bases $B_i \subseteq \mathcal{L}_t^{cd}$ for all participants P_i. The belief base of the analysis agent consists of the information provided by the intake agent, see section 5.7, those of the inquiring agents is initially empty, while the belief bases of the external contacts are blind.

Definition 6.2.8. The **logic** L^{cd} is $ASPIC^+$, the associated inference rules are contained in the rule set $\mathcal{R}^{cd}$.[25]

Definition 6.2.9. A **context** $\mathcal{K}^{cd}$ consists of the axiom premises. No axiom premises are used, so $\mathcal{K}^{cd} = \emptyset$.

Definition 6.2.10. The **set of effect rules** $\mathcal{C}^{cd}$ specifies for each speech act $s_{P_i}(t) \in \mathcal{L}_c$ its effects on the commitments of a participant. C_i^m denotes the commitment set of participant P_i during move m. The following effect rules hold:

1. $m : argue_{P_i 2 P_j}(A) \longrightarrow C_i^m = C_i^{m-1} \cup Prop(A)$
2. $m : report_{P_i 2 P_j}(A) \longrightarrow C_i^m = C_i^{m-1} \cup Prop(A)$
3. $m : concede_{P_i 2 P_j}(A) \longrightarrow C_i^m = C_i^{m-1} \cup Prop(A)$
4. $m : retract_{P_i 2 P_j}(A) \longrightarrow C_i^m = C_i^{m-1} \setminus Prop(A)$

All other utterances have no effect on the commitment base.

[25] See section 6.5

Definition 6.2.11. The **turn-taking function** T^{cd} returns for each dialogue the participant-to-move. The following holds:

1. The analysis agent moves first.
2. After the analysis agent's turn, the first inquiring agent responds. The two agents have a two-person single-move conversation until there are no more responses or follow-ups possible within this sub-dialogue. Then, the turn shifts to the next inquiring agent and this process repeats until all have responded.
3. There are three instances where the protocol is not single-move:
   - The initiator $begin_{P_{ana}2(P_{inq_1},...,P_{inq_m})}(Casefile)$ must be followed up in the same turn.
   - After conceding or affirming, a participant must play another move.
   - When P_{ana} presents its final argument, it must follow this up with the terminator $finish(Casefile)$.

Definition 6.2.12. The **protocol** Pr^{cd} specifies the legal moves at each stage of the dialogue. For all moves m and dialogues d it holds that $m \in Pr(d)$ if and only if all of the following conditions are satisfied:

Part 1: Meta-dialogue protocol

1. If m is played by participant P_i, then it is the turn of P_i.
2. A participant P_i may only play a speech act of type $claim(A)$ or $argue(A)$ if $Prop(A) \in B_i$. Meaning, they cannot state anything that they do not believe.
3. m is not a repetition of an earlier move, meaning that two child locutions to the same utterance cannot be equal. The sole exception to this condition are follow-ups to moves of type $request(\phi)$. This exception gives agents the ability to 'restart' a conversation on a sub-topic that had already been completed unsatisfactorily.
4. m is not a response to a move that has previously had a concession or a retraction as a response.
5. m is not a response to a request or inquiry if said request or inquiry has already been answered with an affirmation or a denial.
6. If m is the initiation of the dialogue, then m is $begin_{P_{ana}2(P_{inq_1},...,P_{inq_m})}(Casefile)$.
7. If m is a child locution to $begin_{P_{ana}2(P_{inq_1},...,P_{inq_m})}(Casefile)$, then m is a follow-up of one of the following types:
   - $request_{P_{ana}2(P_{inq_1},...,P_{inq_m})}(\phi)$
   - $argue_{P_{ana}2(P_{inq_1},...,P_{inq_m})}(A)$ s.t. $Conc(A) = Casefile$

   The analysis agent can do as many requests as it seems fit until it decides it knows enough to draw a final conclusion. For each request, walk through part 2 of the protocol for each inquiring agent $P_{inq_i} \in \{P_{inq_1}, ..., P_{inq_m}\}$.
8. If m is of type $finish_{P_{ana}2(P_{inq_1},...,P_{inq_m})}(Casefile)$, then it terminates the dialogue. This means that the case file has been completed to the satisfaction of the analysis agent.

Part 2: Protocol for sub-dialogue between P_{ana} and P_{inq_i}

1. If m is a child locution to a move of type $request_{P_{ana}2(P_{inq_1},...,P_{inq_m})}(\phi)$, then m is a follow-up of one of the following types:
   - $affirm_{P_{inq_i}2P_{ana}}(\phi)$
   - $deny_{P_{inq_i}2P_{ana}}(\phi)$

Here, an inquiring agent responds to a request by affirming or denying that its external contact can provide any information on ϕ.

2. The investigation of ϕ by P_{inq_i} is a sub-sub-dialogue that is initiated by $begin_{P_{inq_i}2P_{ext_i}}(information(\phi))$ and terminated by $finish_{P_{inq_i}2P_{ext_i}}(information(\phi))$. We call this the external exchange, see part 3 for this protocol. After termination, the current protocol resumes.

3. If m is a child locution to a move of type $affirm_{P_{inq_i}2P_{ana}}(\phi)$, then m is a follow-up of one of the following types:

   - $challenge_{P_{inq_i}2P_{ana}}(relevant(\phi))$
   - $begin_{P_{inq_i}2P_{ext_i}}(information(\phi))$

   If the inquiring agent requires justification for its inquiry to its external contact, it can play devil's advocate and challenge the relevance of ϕ. This is not the case, for instance, when its contact is located within the police organisation. Once the inquiring agent is ready to start a sub-dialogue with its external contact, it starts with an initiating speech act. Go to part 3 for this protocol.

4. If m is a child locution to a move of type $challenge_{P_{inq_i}2P_{ana}}(relevant(\phi))$, then m is a response of one of the following types:

   - $argue_{P_{ana}2P_{inq_i}}(A)$ s.t. $Conc(A) = relevant(\phi)$
   - $retract_{P_{ana}2P_{inq_i}}(\phi)$

   The analysis agent can respond to a challenge by an inquiring agent with either an argument or a retraction.

5. If m is a child locution to a move of type $argue_{P_{ana}2P_{inq_i}}(A)$, then m is a response of one of the following types:

   - $challenge_{P_{inq_i}2P_{ana}}(\psi)$ where $\psi \in VulProp(A) \setminus Conc(A)$
   - $concede_{P_{inq_i}2P_{ana}}(A)$

   The inquiring agent can press for more information or concede the argument given by the analysis agent.

6. If m is a child locution to a move of type $finish_{P_{inq_i}2P_{ext_i}}(information(\phi))$, then m is a follow-up of type $report_{P_{inq_i}2P_{ana}}(A)$ where $Conc(A)$ is $information(\phi)$ or $\neg information(\phi)$. Here, the inquiring agent reports what it has learned in the external exchange. If the contact has given any information, then this is constructed as an argument with $information(\phi)$ as its conclusion. If not, the resulting argument is simply $\neg information(\phi)$.

7. If m is a child locution to a move of type $report_{P_{inq_i}2P_{ana}}(A)$, then m is a follow-up of type $concede_{P_{ana}2P_{inq_i}}(A)$. An analysis agent cannot attack or challenge the information provided by an inquiring agent.

8. m is not a child locution to a move of one of the following types:

   - $concede_{P_{inq_i}2P_{ana}}(A)$
   - $concede_{P_{ana}2P_{inq_i}}(A)$
   - $retract_{P_{ana}2P_{inq_i}}(\phi)$
   - $deny_{P_{inq_i}2P_{ana}}(\phi)$

Part 3: Protocol for external exchange between P_{inq_i} and P_{ext_i}

1. If m is a child locution to a move of type $begin_{P_{inq_i}2P_{ext_i}}(information(\phi))$, then m is a follow-up of type $inquire_{P_{inq_i}2P_{ext_i}}(\phi)$. The inquiring agent can only discuss the subject that the analysis agent has brought up earlier.

2. If m is a child locution to a move of type $inquire_{P_{inq_i}2P_{ext_i}}(\phi)$, then m is a response of one of the following types:

- $affirm_{P_{ext_i}2P_{inq_i}}(\phi)$
- $deny_{P_{ext_i}2P_{inq_i}}(\phi)$

The external contact states whether or not it is able to provide any information about ϕ.

3. If m is a child locution to a move of type $affirm_{P_{ext_i}2P_{inq_i}}(\phi)$, then m is a follow-up of one of the following types:

   - $challenge_{P_{ext_i}2P_{inq_i}}(relevant(\phi))$
   - $report_{P_{ext_i}2P_{inq_i}}(A)$ s.t. $Conc(A) = information(\phi)$

   The external contact can choose to ask for a justification for providing the requested information, or to give it right away.

4. m is not a response to a move of type $affirm_{P_{ext_i}2P_{inq_i}}(\phi)$ that has an earlier response of type $report_{P_{ext_i}2P_{inq_i}}(A)$ s.t. $Conc(A) = information(\phi)$. Meaning, that once the external contact has given up the relevant information, it cannot ask for a explanation afterwards.

5. If m is a child locution to a move of type $challenge_{P_{ext_i}2P_{inq_i}}(relevant(\phi))$, then m is a response of one of the following types:

   - $argue_{P_{inq_i}2P_{ext_i}}(A)$ s.t. $Conc(A) = relevant(\phi)$
   - $retract_{P_{inq_i}2P_{ext_i}}(\phi)$

   The inquiring agent can give an argument in favour of providing the relevant information, or retract its request.

6. If m is a child locution to a move of type $argue_{P_{inq_i}2P_{ext_i}}(A)$, then m is a response of one of the following types:

   - $challenge_{P_{ext_i}2P_{inq_i}}(\psi)$ where $\psi \in VulProp(A) \setminus Conc(A)$
   - $concede_{P_{ext_i}2P_{inq_i}}(A)$

   The external contact can press further or accept the argument provided by the inquiring agent.

7. If m is a child locution to a move of type $report_{P_{ext_i}2P_{inq_i}}(A)$, then m is a response of type $concede_{P_{inq_i}2P_{ext_i}}(A)$. An inquiring agent cannot attack or challenge the information provided by the external source.

8. If m is a child locution to a move of type $concede_{P_{inq_i}2P_{ext_i}}(A)$, then m is a follow-up of type $finish_{P_{inq_i}2P_{ext_i}}(information(\phi))$. Conceding the information terminates the sub-dialogue.

9. If m is a child locution to a move of type $deny_{P_{ext_i}2P_{inq_i}}(\phi)$, then m is a response of type $finish_{P_{inq_i}2P_{ext_i}}(information(\phi))$. If the external contact denies having the ability to provide information, then the sub-dialogue terminates.

10. m is not a child locution to a move of one of the following types:

    - $concede_{P_{ext_i}2P_{inq_i}}(A)$
    - $retract_{P_{inq_i}2P_{ext_i}}(\phi)$

Definition 6.2.13. The single **outcome rule** $o_1 \in O^{cd}$ states that the outcome of the dialogue is the final argument the analysis agent can form based on the investigation, with $Casefile$ as its conclusion.

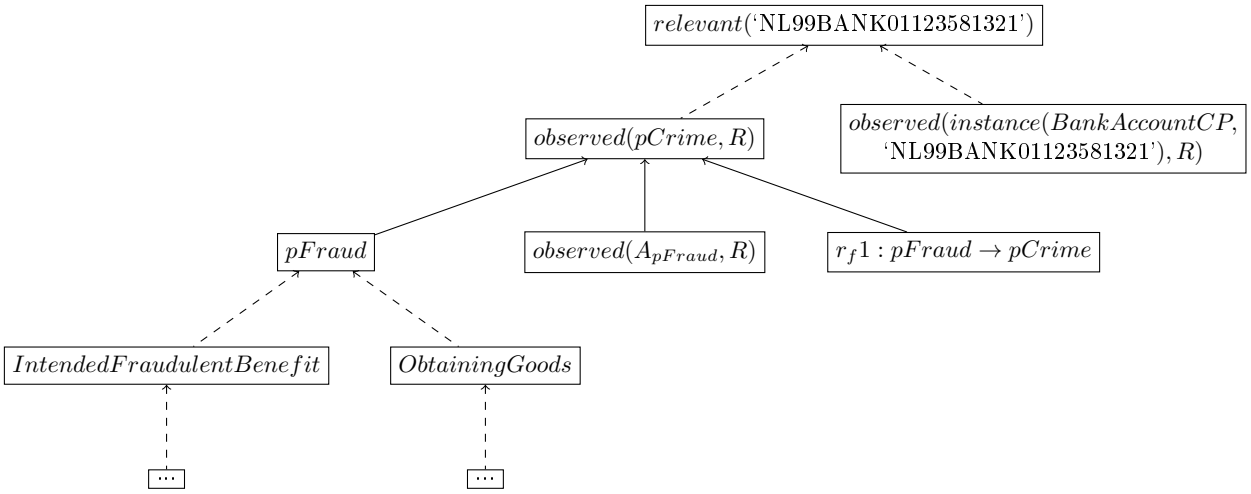

Figure 13: An argument for affirming relevance of topic ϕ.

6.3 Forming an argument of relevance

In the protocol, we stated that an agent may be asked to elaborate on the relevance of inquiring information about a certain subject. How this argument is to be constructed, depends on the input given by the intake agent to the analysis agent prior to the dialogue. We gave an informal example of such an argument in Figure 2 earlier. We introduce the following two rules in our case file rule set $\mathcal{R}_{cf}$:

Rules				**Preference**
$r_{cf}1$	$observed(pCrime, R), observed(instance(\chi, \phi), R)$	$\Rightarrow$	$relevant(\phi)$	High
$r_{cf}2$	$A, observed(A, R), r : (Conc(A) \rightarrow \phi)$	$\rightarrow$	$observed(\phi, R)$	High

Table 9: Rule set $\mathcal{R}_{cf}$, version 1.

The philosophy behind $r_{cf}1$ is this: if something has been observed in a police report that presumably concerns a crime, then it is relevant. $r_{cf}2$ states the following:

1. Given a police report R in the topic language $\mathcal{L}_t^{cd}$, which is an argument that has been provided by the intake agent through the function q,[26]
2. as well as the sub-argument A in R, which must necessarily also be part of the topic language under q,
3. along with a strict rule r with $Conc(A)$ as its antecedent,
4. we then assume that the consequent of r is also observed in R.

Remember that under rule set $\mathcal{R}_f$, we have the strict rule $r_f1 : pFraud \rightarrow pCrime$.[27] So, for an argument A that has $pFraud$ as its conclusion and that has been observed in a police report R (any successful police report should have such an argument), we may conclude that $pCrime$ is also observed in the same police report. Now, this might sound like an unnecessary procedure, since we can already conclude $observed(pCrime, R)$ from R alone, but it is predominantly a way to append an argument from a police report into an argument of relevance. See figure 13 for an illustration of this. $r_{cf}2$ ensures that the analysis agent can only cite an argument if it is directly related to the topic of relevance, i.e. if both are mentioned in the same police report.

6.4 Providing information

The external contacts, when given a legitimate reason to, are expected to provide the information that is at their disposal surrounding a certain topic. This information is presented in the form of

[26] Section 5.7.
[27] See Table 4.

an argument. We expand $\mathcal{R}_{cf}$ with an additional rule concerning the passing along of information.

Rules				Preference
$r_{cf}1$	$observed(pCrime, R), observed(instance(\chi, \phi), R)$	$\Rightarrow$	$relevant(\phi)$	High
$r_{cf}2$	$A, observed(A, R), r : (Conc(A) \rightarrow \phi)$	$\rightarrow$	$observed(\phi, R)$	High
$r_{cf}3$	$check(NameRegistered)$, $check(AddressRegistered)$, $check(ResidenceRegistered)$, $check(EmailAddressRegistered)$, $check(PhoneNumberRegistered)$, $check(BankAccountRegistered)$, $check(IPAddressRegistered)$	$\rightarrow$	$information(\phi)$	High

Table 10: Rule set $\mathcal{R}_{cf}$, version 2.

$r_{cf}3$ requires the earlier rule set $\mathcal{R}_i$ to make sense, specifically r_i5, r_i6 and r_i7.[28] Not every external contact can be expected to be able to provide the name, the address, the residence, the e-mail address, the phone number, the bank account and the IP address of the suspect. However, they should state whether or not they have the information at their disposal, through the use of the $check()$-type of proposition. We assume that the strategy of the external contacts is to fully disclose all information they possess once this has been demanded, so that an additional $asktospecify()$-move is not needed in this situation.

Between the investigation and the final conclusion, *post-investigation matching* can take place.[29] This can tie information from earlier case file dialogues to the current case. If an instance such as $NameRegistered$ occurs in two separate case file processes, the analysis agent's belief base from the earlier dialogue can be added to the belief base in the later one.

6.5 Forming a concluding argument

After the investigation phase, the analysis agent needs to report on the identity and modus operandi of the suspect. Given that the suspect might have used false information to obtain their goods, it is likely that some intake data deviates from the investigation data. Generally speaking, the external sources are more reliable than the information given by the applicant. This should be reflected in the argumentation rules. We expand $\mathcal{R}_{cf}$ once more, this time by an additional 33 rules, see Table 11.

Now that $\mathcal{R}_{cf}$ has been finalized, we can formally define the argumentation system for case file dialogues AS^{cd}. All rules that have been defined throughout this thesis, are necessary for the case file dialogue.

Definition 6.5.1. Given rule set $\mathcal{R}^{cd} = \{\mathcal{R}^{id} \cup \mathcal{R}_{cf} \cup \mathcal{R}_{cf}\}$[30]$= \{\mathcal{R}_s^{cd} \cup \mathcal{R}_d^{cd}\}$[31], topic language $\mathcal{L}_t^{id}$ and partial function $n^{cd} : R_d^{cd} \longrightarrow \mathcal{L}^{cd}$, we have an argumentation system AS^{cd} for case file dialogues.

We will give a brief explanation of the new rules. For each of the seven topics (name, address, residence, e-mail address, phone number, bank account number and IP address) the analysis agent may or may not receive information from their external contacts. This information is seen as more reliable than what has been stated about the same topic by the applicants (compare the preferences of $r_{cf}4$ and $r_{cf}5$, for example). If both sources (applicant and external) state different things, then r_i10[32], which allows for the coexistence of two instances, does not hold for these specific instances (see $r_{cf}6$, for example). The claims of the applicants can be seen as indicators for a false identity or money laundering (see $r_{cf}7$, $r_{cf}14$, $r_{cf}21$ and $r_{cf}33$). This does not apply for divergent information from two different external contacts (see $r_{cf}39$). The identity ($r_{cf}41$), false identity ($r_{cf}42$) and crime description from the police reports together form the case file argument $A_{Casefile}$ (see $r_{cf}43$).

[28] See Table 5.
[29] See section 5.7.
[30] See Tables 4, 5, 6, 8 and 11.
[31] $\mathcal{R}_s^{cd}$ and $\mathcal{R}_d^{cd}$ denote the strict and defeasible rule subsets of $\mathcal{R}^{id}$, respectively (see Definition 4.1.5).
[32] See Table 5.

The argument $A_{Casefile}$, constructed by the analysis agent based on the information obtained from the intake agent or through the dialogue, should:

1. have only observables and specifications of observables as their premises (if the specifications are known, they must be included),
2. be justified according to B_{int} under the weakest-link principle,
3. have $Casefile$ as its conclusion,
4. not be less preferred than other possible arguments that meet the previous three requirements according to B_{int} under the weakest-link principle.

The first three requirements mirror those in section 5.5. The set of observables Ω^{cd} in $\mathcal{L}_t^{cd}$ contains the same observables as Ω^{id} (see Figure 4). $instance(\phi, \psi)$ counts as a specification of the observable $known(\phi)$, even if the latter is not part of the argument. The fourth requirement is added because of the preference relation in $r_{cf}32$: the more observations are added to the *Identity* sub-argument, the higher the preference of the entire argument $A_{Casefile}$ under the weakest-link principle. This stimulates extensive arguments.

The sub-argument of $A_{Casefile}$ with $pCrime$ as its conclusion (see $r_{cf}43$) can be one of two things: either the argument A_{pCrime} taken directly from just one of the submitted reports, as we could argue that matched cases will have similar crime descriptions, or a new argument composed of the A_{pCrime} arguments of *all* matched reports. For example, if R_1 has a sub-argument '$ContactOverPhone \Rightarrow ContactPrior$', and R_2 has a sub-argument '$ContactOverEmail \Rightarrow ContactPrior$', then '$ContactOverPhone, ContactPrior \Rightarrow \phi$' is a subargument of $A_{Casefile}$ for $\{R_1, R_2\}$. We choose to use the compositional argument instead of a random argument from a single report, since this will give the most complete overview of the case.

6.6 Feedback to intake agent

It is possible that at some point in the investigation, it has been established that a specific case does not involve fraud, even though the reports suggest so. This could for instance be indicated by whitelisting specific trade websites or salespersons, or through human intervention. The analysis agent should be able to communicate this to the intake agent, in order to stop it from gathering any further reports about the same suspect. One possible way to do this by introducing an additional rule r_0 that states that a specific instance of a category negates the fraud claim: $r_0 : instance(\psi, \phi) \Rightarrow \neg pFraud$ for a predetermined ϕ and ψ, where $r_f 2 \prec r_0$.

6.7 Example

Say that the fictional counterparty in the example of the intake dialogue has committed a couple of similar crimes, using the same name, e-mail address and bank account. The example in Figures 14 and 15 shows how the subsequent case file dialogue could play out.

Rules				Preference
$r_{cf}1$	$observed(pCrime, R)$,			
	$observed(instance(\chi, \phi), R)$	$\Rightarrow$	$relevant(\phi)$	High
$r_{cf}2$	$A, observed(A, R), r : (Conc(A) \rightarrow \phi)$	$\rightarrow$	$observed(\phi, R)$	High
$r_{cf}3$	$check(NameRegistered)$,			
	$check(AddressRegistered)$,			
	$check(ResidenceRegistered)$,			
	$check(EmailAddressRegistered)$,			
	$check(PhoneNumberRegistered)$			
	$check(BankAccountRegistered)$			
	$check(IPAddressRegistered)$	$\rightarrow$	$information(\phi)$	High
$r_{cf}4$	$instance(NameRegistered, \phi)$	$\Rightarrow$	$instance(NameSuspect, \phi)$	High
$r_{cf}5$	$instance(NameCP, \phi)$	$\Rightarrow$	$instance(NameSuspect, \phi)$	Medium
$r_{cf}6$	$instance(NameRegistered, \chi)$,			
	$instance(NameCP, \psi)$	$\Rightarrow$	$\neg r_i 10$ (for $\phi = NameSuspect$, same χ, ψ)	High
$r_{cf}7$	$instance(NameRegistered, \chi)$,			
	$instance(NameCP, \psi)$	$\Rightarrow$	$instance(FakeName, \psi)$	High
$r_{cf}8$	$\neg known(NameRegistered)$	$\Rightarrow$	$\neg known(NameSuspect)$	Low
$r_{cf}9$	$\neg known(NameCP)$	$\Rightarrow$	$\neg known(FakeName)$	Low
$r_{cf}10$	$instance(NameRegistered, \phi)$,			
	$instance(NameCP, \phi)$	$\Rightarrow$	$\neg known(FakeName)$	Low
$r_{cf}11$	$instance(AddressRegistered, \phi)$	$\Rightarrow$	$instance(AddressSuspect, \phi)$	High
$r_{cf}12$	$instance(AddressCP, \phi)$	$\Rightarrow$	$instance(AddressSuspect, \phi)$	Medium
$r_{cf}13$	$instance(AddressRegistered, \chi)$,			
	$instance(AddressCP, \psi)$	$\Rightarrow$	$\neg r_i 10$ (for $\phi = AddressSuspect$, same χ, ψ)	High
$r_{cf}14$	$instance(AddressRegistered, \chi)$,			
	$instance(AddressCP, \psi)$	$\Rightarrow$	$instance(FakeAddress, \psi)$	High
$r_{cf}15$	$\neg known(AddressRegistered)$	$\Rightarrow$	$\neg known(AddressSuspect)$	Low
$r_{cf}16$	$\neg known(AddressCP)$	$\Rightarrow$	$\neg known(FakeAddress)$	Low
$r_{cf}17$	$instance(AddressRegistered, \phi)$,			
	$instance(AddressCP, \phi)$	$\Rightarrow$	$\neg known(FakeAddress)$	Low
$r_{cf}18$	$instance(ResidenceRegistered, \phi)$	$\Rightarrow$	$instance(ResidenceSuspect, \phi)$	High
$r_{cf}19$	$instance(ResidenceCP, \phi)$	$\Rightarrow$	$instance(ResidenceSuspect, \phi)$	Medium
$r_{cf}20$	$instance(ResidenceRegistered, \chi)$,			
	$instance(ResidenceCP, \psi)$	$\Rightarrow$	$\neg r_i 10$ (for $\phi = ResidenceSuspect$, same χ, ψ)	High
$r_{cf}21$	$instance(ResidenceRegistered, \chi)$,			
	$instance(ResidenceCP, \psi)$	$\Rightarrow$	$instance(FakeResidence, \psi)$	High
$r_{cf}22$	$\neg known(ResidenceRegistered)$	$\Rightarrow$	$\neg known(ResidenceSuspect)$	Low
$r_{cf}23$	$\neg known(ResidenceCP)$	$\Rightarrow$	$\neg known(FakeResidence)$	Low
$r_{cf}24$	$instance(ResidenceRegistered, \phi)$,			
	$instance(ResidenceCP, \phi)$	$\Rightarrow$	$\neg known(FakeResidence)$	Low
$r_{cf}25$	$instance(EmailAddressRegistered, \phi)$	$\Rightarrow$	$instance(EmailAddressSuspect, \phi)$	High
$r_{cf}26$	$nstance(EmailAddressCP, \phi)$	$\Rightarrow$	$instance(EmailAddressSuspect, \phi)$	High
$r_{cf}27$	$\neg known(EmailAddressRegistered)$	$\Rightarrow$	$\neg known(EmailAddressSuspect)$	Low
$r_{cf}28$	$instance(PhoneNumberRegistered, \phi)$	$\Rightarrow$	$instance(PhoneNumberSuspect, \phi)$	High
$r_{cf}29$	$instance(PhoneNumberCP, \phi)$	$\Rightarrow$	$instance(PhoneNumberSuspect, \phi)$	High
$r_{cf}30$	$\neg known(PhoneNumberRegistered)$	$\Rightarrow$	$\neg known(PhoneNumberSuspect)$	Low
$r_{cf}31$	$instance(BankAccountRegistered, \phi)$	$\Rightarrow$	$instance(BankAccountSuspect, \phi)$	High
$r_{cf}32$	$instance(BankAccountCP, \phi)$	$\Rightarrow$	$instance(BankAccountSuspect, \phi)$	Medium
$r_{cf}33$	$instance(BankAccountRegistered, \chi)$,			
	$instance(BankAccountCP, \psi)$	$\Rightarrow$	$instance(MoneyLaundering, \psi)$	High
$r_{cf}34$	$\neg known(BankAccountRegistered)$	$\Rightarrow$	$\neg known(BankAccountSuspect)$	Low
$r_{cf}35$	$\neg known(BankAccountCP)$	$\Rightarrow$	$\neg known(MoneyLaundering)$	Low
$r_{cf}36$	$instance(BankAccountRegistered, \phi)$,			
	$instance(BankAccountCP, \phi)$	$\Rightarrow$	$\neg known(MoneyLaundering)$	Low
$r_{cf}37$	$instance(IPAddressRegistered, \phi)$	$\Rightarrow$	$instance(IPAddressSuspect, \phi)$	High
$r_{cf}38$	$\neg known(IPAddressRegistered)$	$\Rightarrow$	$\neg known(IPAddressSuspect)$	Low
$r_{cf}39$	$instance(\phi, \psi_1), ..., instance(\phi, \psi_n)$	$\Rightarrow$	$identity(\phi, \{\psi_1 \vee ... \vee \psi_n\})$	High (increase with n)
$r_{cf}40$	$\neg known(\phi)$	$\Rightarrow$	$identity(\phi, \emptyset)$	Low
$r_{cf}41$	$identity(NameSuspect, \alpha)$,			
	$identity(AddressSuspect, \beta)$,			
	$identity(ResidenceSuspect, \gamma)$,			
	$identity(EmailAddressSuspect, \delta)$,			
	$identity(PhoneNumberSuspect, \epsilon)$,			
	$identity(BankAccountSuspect, \zeta)$,			
	$identity(IPAddressSuspect, \eta)$,	$\Rightarrow$	$Identity$	High
$r_{cf}42$	$identity(FakeName, \alpha)$,			
	$identity(FakeAddress, \beta)$,			
	$identity(FakeResidence, \gamma)$,			
	$identity(MoneyLaundring, \delta)$,	$\Rightarrow$	$FalseIdentity$	High
$r_{cf}43$	$pCrime, Identity, FalseIdentity$	$\rightarrow$	$Casefile$	

Table 11: Rule set $\mathcal{R}_{cf}$, version 3.

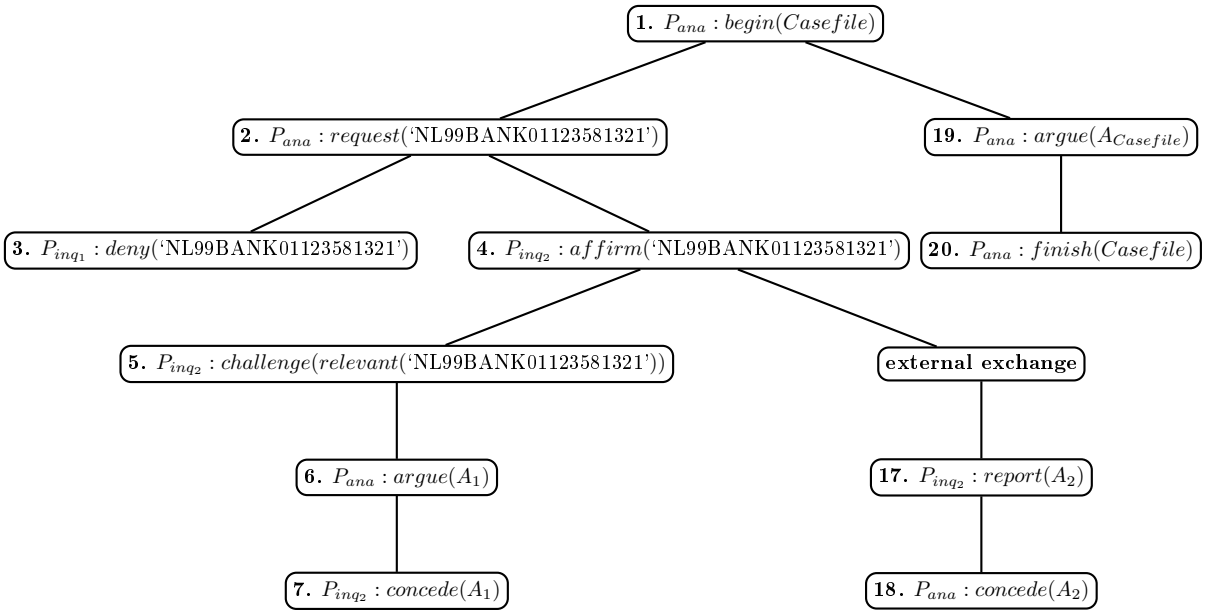

Figure 14: An example of a case file dialogue in which the analysis agent wants to obtain information about the bank account number NL99BANK01123581321. This tree shows the dialogue between the analysis agent and two inquiring agents.

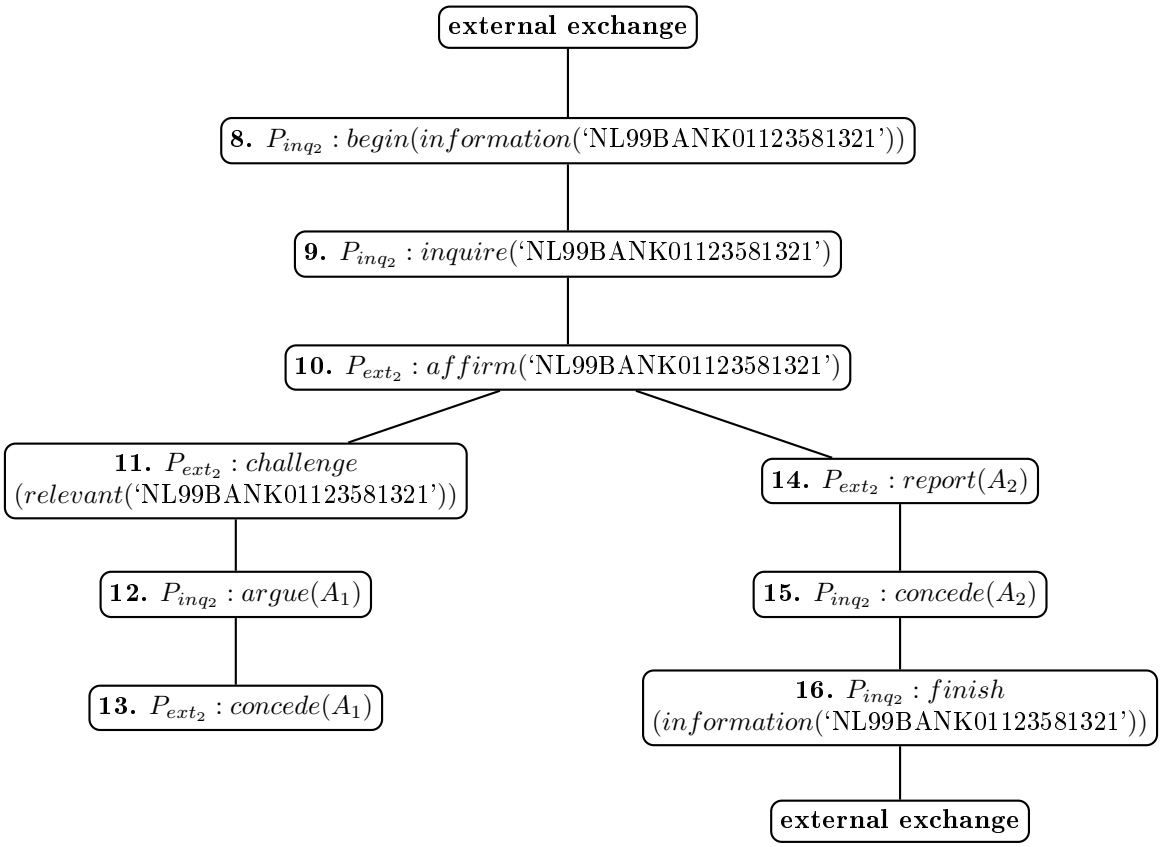

Figure 15: An example of a case file dialogue in which the analysis agent wants to obtain information about the bank account number NL99BANK01123581321. This tree shows the dialogue between an inquiring agent and its external contact.

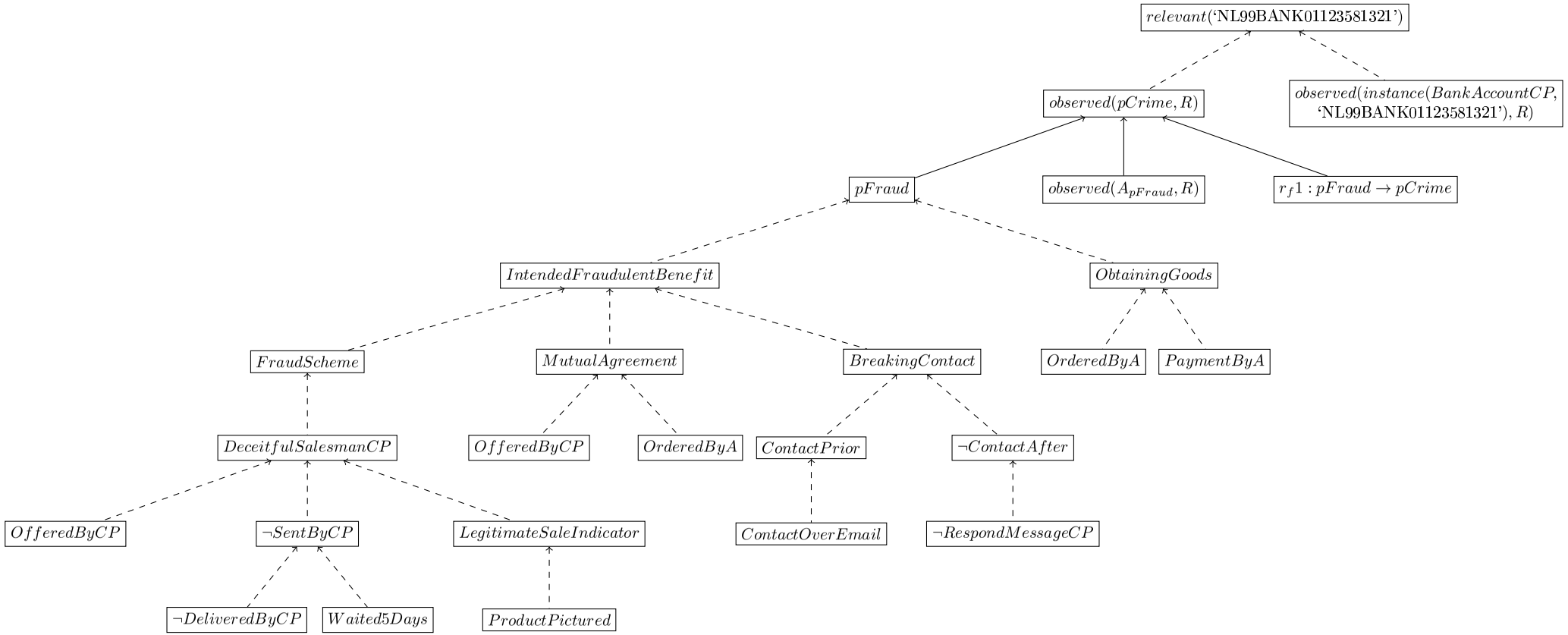

Figure 16: The argument A_1 referred to in Figures 14 and 15.

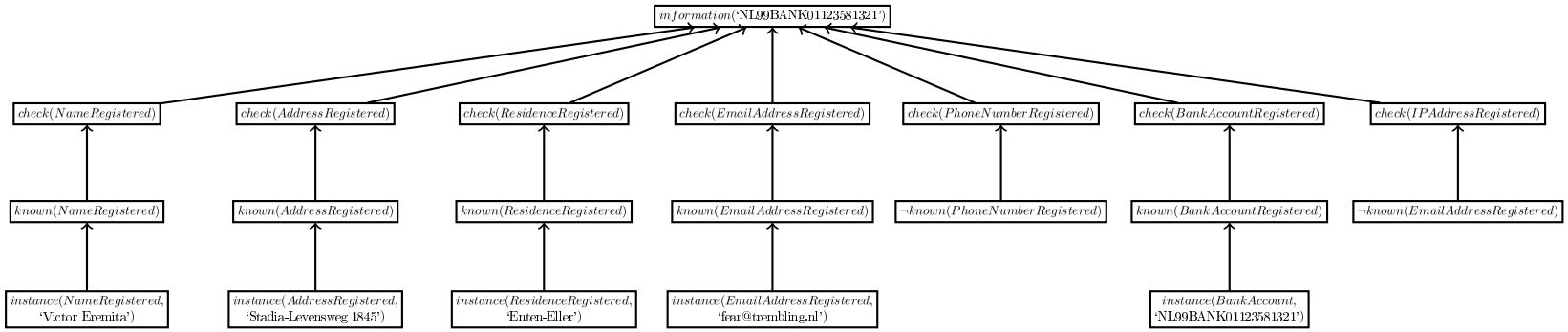

Figure 17: The argument A_1 referred to in Figures 14 and 15.

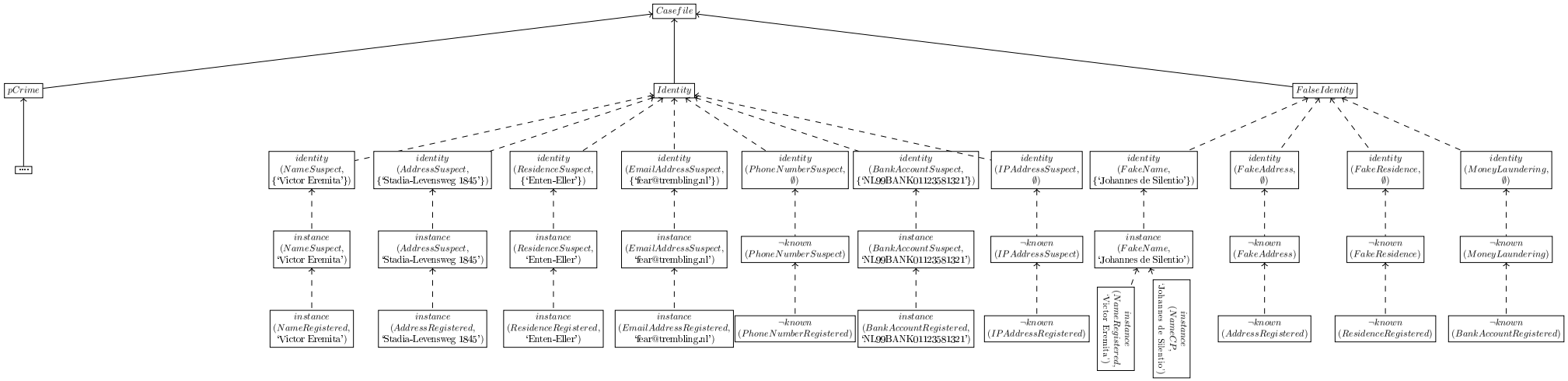

Figure 18: The concluding argument $A_{Casefile}$ referred to in Figure 14. The argument for $pCrime$ is identical to Figure 10.

7 Case study

In this case study, we apply the defined dialogue systems to data from existing police reports. For reasons of privacy, all identifiable data has been changed. Since the original reports are in Dutch, the data had to be translated into English. We have intended to remain as faithful to the original texts as possible, such that a hypothetical perfect NER parser would extract the same observations in both languages. The four selected reports all point to the same suspect, making them a useful starting point for illustrating the complete process that starts with the intake dialogue and ends with a case file.

7.1 Intake data

The original intake data was obtained from a CSV file containing the answers applicants had given to the questions in Tables 2 and 3, as well as whether or not the report had been withdrawn and for what reason, and whether or not a specific follow-up question had been asked and answered. The translation of the obtained information can be found in Table 12 and Figure 19. None of the four reports were later withdrawn or resulted in any follow-up questions.

Intake number	1.	2.	3.	4.
Name A	Applicant 1	Applicant 1	Applicant 2	Applicant 3
Phone number A	06-1111	06-1111	06-2222	020-3333
Mobile phone number A	06-1111	06-1111	06-2222	06-3333
Name CP	Dennis de Boer	Dennis de Boer	Lukas Goedeman	-
Username CP	Toys4ever	Toys4ever	Toys4ever	Toys4ever
E-mail address CP	-	-	-	-
Zip code CP	-	-	-	-
House number CP	-	-	-	-
House number suffix CP	-	-	-	-
Street name CP	-	-	-	-
Residence CP	Volendam	Volendam	-	-
Country CP	Netherlands	Netherlands	Netherlands	Netherlands
Phone number CP	-	-	-	-
Mobile phone number CP	-	-	-	-
Trade website	marktplaats.nl	marktplaats.nl	marktplaats.nl	marktplaats.nl
Advertisement title	Barbie doll	Dikkie Dik books	Barbie doll	-
Advertisement number	m0001	m0002	m0001	m0002
Additional information	-	-	-	-
Transaction date	08-10-2017	08-10-2017	09-10-2017	06-10-2017
Purchase money	30	41.95	33.95	126.95
Method of payment	Bank transfer (IBAN)	Bank transfer (IBAN)	iDEAL (IBAN)	Bank transfer (IBAN)
Bank account number CP	NL00BANK1234	NL00BANK1234	NL00BANK1234	NL00BANK1234
Account holder CP	Dennis de Boer	Dennis de Boer	Lukas Goedeman	Lukas Goedeman
Paypal e-mail address CP	-	-	-	-
Through trade website	-	-	-	-
Permission to contact CP	TRUE	TRUE	TRUE	TRUE
Social media	-	-	-	-
Transaction time	11:22	11:22	-	17:45
Bank account number A	NL00GELD7777	NL00GELD7777	NL00CASH9999	NL00MUNT6666

Table 12: Anonymized and translated intake data of four raw police reports, excluding conflict descriptions.

7.2 Intake dialogues

We have worked out possible intake dialogues based on the four reports, given that the dialogue system is $\mathcal{D}^{id}$ and the argumentation system is AS^{id}.[33] We assume $B^0_{int} = \emptyset$. The full dialogues can be read in Appendix B, we will give a general description below.

7.2.1 Intake 1

The first intake starts with a number of concessions by the intake agent to the claims of the applicant about the conflict description. What is missing, is whether or not the applicant has

[33] See definitions 5.2.1 and 5.3.1.

1. "On 08/10, I ordered two toy items sold by Toys4ever through Marktplaats. One of these items was a Barbie doll, which was offered in the aforementioned advertisement. Together with the other product (see next report), this item would be sent on the same day, such that it would be delivered on 10/10. That is why I transferred the amount of 71,95 to the aforementioned bank account, which appeared to belong to Lukas Goedeman from Zwijndrecht. However, the ordered items were not delivered and the salesperson does not respond to my messages. Furthermore, he promised me three times he would give me a PostNL Track&Trace code, but this he also did not do."
2. "On *08/10, I ordered two toy items sold by Toys4ever through Marktplaats. One of these products was a Dikkie Dik book series (eight-part), which were offered in the aforementioned advertisement. Together with the other product (see previous report), this item would be sent on the same day, such that it would be delivered on 10/10. That is why I transferred the amount of 71,95 to the bank account NL00BANK1234, that according to the salesperson belonged to Lukas Goedeman from Zwijndrecht. The transferred amount consists of 35 Euro for the books and 30 Euro for the doll, plus 6,95 Euro shipping costs. The ordered items were not delivered and the salesperson does not respond to my messages. Furthermore, he promised me three times he would give me a PostNL Track&Trace code, but this he also did not do. In the meantime, I have discovered through the bank of the account holder that the account belongs to Dennis de Boer, and not to Lukas Goedeman"
3. "I had contact with Lukas Goedeman (if that is his real name...) through Marktplaats. He claimed to have a Barbie doll for sale. He gave me his details and an article number. He appeared to be active on Marktplaats for 5.5 years. The asking price was 30 Euro and the shipping costs were 3.95. I transferred the amount to NL00BANK1234 in the name of Lukas Goedeman in Zwijndrecht. The advertisement was subsequently deleted and after that, I have not heard anything from him. I contacted Marktplaats and they told me they had already received multiple complaints and they advised me to file a police report."
4. "On 06/10 at 16:37, I responded to the aforementioned advertisement (11 Dikkie Dik books) and offered €120. After the selling party agreed to this, I got an e-mail at 17:12 containing the bank account number and the promise that the books would be sent and that I would receive a Track & Trace code. The same evening, I transferred the amount to NL00BANK1234 in the name of Lukas Goedeman in Zwijndrecht. The last response I got was on 06/10 at 17:53. Despite multiple attempts, I have to this day not heard anything from the selling party."

Figure 19: Anonymized and translated conflict descriptions of the reports in table 12.

waited five days for their product to be delivered before submitting the report and if/how they had had prior contact with the counterparty. We assume they have waited at least five days, and that all contact went through Marktplaats (since no e-mail address is mentioned). The specification exchange is also mostly a claim and concession back-and-forth between the intake agent and the applicant. Something unique about the report is the fact that the applicant has done some investigation of their own (see Intake 2) and concluded that the name of the bank account holder (Dennis de Boer) is not the name stated by the salesman (Lukas Goedeman). Under the applied intake protocol, either $instance(AccountHolderCP, \text{'Dennis de Boer'})$ or $instance(AccountHolderCP, \text{'Lukas Goedeman'})$ should be retracted, unless the applicant argues that both are true. We imagine the applicant retracting $instance(AccountHolderCP, \text{'Lukas Goedeman'})$, since they state in the conflict description of Intake 2 that 'Lukas Goedeman' is not the account holder. The concluding report of Intake 1 will be referred to as R_1.

7.2.2 Intake 2

The second intake is in many ways similar to the previous intake, since it has the same applicant and the same counterparty, although it concerns a different product and price. Furthermore, the conflict description is a little more extensive, including an argument why 'Lukas Goedeman' is not the account holders name. The concluding report of Intake 2 will be referred to as R_2.

Intake 1:

- $OrderedByA$
- $OfferedByCP$
- $PaymentByA$
- $\neg DeliveredByCP$
- $\neg RespondMessageCP$
- $TrackAndTrace$
- $instance(NameA, \text{'Applicant 1'})$
- $instance(PhoneNumberA, \text{'06-1111'})$
- $instance(MobilePhoneA, \text{'06-1111'})$
- $instance(NameCP, \text{'Dennis de Boer'})$
- $instance(UsernameCP, \text{'Toys4Ever'})$
- $instance(ResidenceCP, \text{'Heerenveen'})$
- $instance(CountryCP, \text{'Netherlands'})$
- $instance(TradeWebsite, \text{'marktplaats.nl'})$
- $instance(AdvertisementTitle, \text{'Barbie doll'})$
- $instance(AdvertisementNumber, \text{'m0001'})$
- $instance(TransactionDate, \text{'08-10-2017'})$
- $instance(PurchaseMoney, \text{'€30'})$
- $instance(PaymentMethod, \text{'Bank transfer (IBAN)'})$
- $instance(BankAccountCP, \text{'NL00BANK1234'})$
- $instance(AccountHolderCP, \text{'Dennis de Boer'})$
- $instance(AccountHolderCP, \text{'Lukas Goedeman'})$
- $instance(TransactionTime, \text{'11:22'})$
- $instance(BankAccountA, \text{'NL00GELD7777'})$
- $instance(ProductType, \text{'Barbie doll'})$

Intake 2:

- $OrderedByA$
- $OfferedByCP$
- $PaymentByA$
- $\neg DeliveredByCP$
- $\neg RespondMessageCP$
- $TrackAndTrace$
- $instance(NameA, \text{'Applicant 1'})$
- $instance(PhoneNumberA, \text{'06-1111'})$
- $instance(MobilePhoneA, \text{'06-1111'})$
- $instance(NameCP, \text{'Dennis de Boer'})$
- $instance(UsernameCP, \text{'Toys4ever'})$
- $instance(ResidenceCP, \text{'Volendam'})$
- $instance(CountryCP, \text{'Netherlands'})$
- $instance(TradeWebsite, \text{'marktplaats.nl'})$
- $instance(AdvertisementTitle, \text{'Dikkie Dik books'})$
- $instance(AdvertisementNumber, \text{'m0002'})$
- $instance(TransactionDate, \text{'08-10-2017'})$
- $instance(PurchaseMoney, \text{'€41,95'})$
- $instance(PaymentMethod, \text{'Bank transfer (IBAN)'})$
- $instance(BankAccountCP, \text{'NL00BANK1234'})$
- $instance(AccountHolderCP, \text{'Dennis de Boer'}) \Rightarrow \neg instance(AccountHolderCP, \text{'Lukas Goedeman'})$
- $instance(TransactionTime, \text{'11:22'})$
- $instance(BankAccountA, \text{'NL00GELD7777'})$
- $instance(ProductType, \text{'Dikkie Dik book series (8-part)'})$

Figure 20: Observations based on Intake 1 & 2

7.2.3 Intake 3

The third intake leaves some information implicit, such as the fact that no product was delivered. The intake agent should ask whether or not this is indeed the case, in order to make a conclusion about $pCrime$. Other missing information is the waiting period, the method of contact, the legitimate sale indicator, how the contact was broken, and the transaction time. The concluding report of Intake 3 will be referred to as R_3.

7.2.4 Intake 4

The fourth intake, strangely enough, omits the counterparty's name, even though 'Lukas Goedeman' is stated as the name of the bank account holder. Perhaps the applicant suspected that it was either a fake name or that the bank account holder and the salesman were not the same person. In any case, $NameCP$ is to be inquired by the intake agent and the obvious answer is $instance(NameCP, \text{Lukas Goedeman})$. Also missing in the report is the waiting period, whether there was a delivery, if there was prior contact, and how the contact was broken afterwards. The concluding report of Intake 4 will be referred to as R_4.

Intake 3:

- $OrderedByA$
- $OfferedByCP$
- $PaymentByA$
- $\neg ContactAfter$
- $instance(NameA, \text{'Applicant 2'})$
- $instance(PhoneNumberA, \text{'06-2222'})$
- $instance(MobilePhoneA, \text{'06-2222'})$
- $instance(NameCP, \text{'Lukas Goedeman'})$
- $instance(UsernameCP, \text{'Toys4ever'})$
- $instance(CountryCP, \text{'Netherlands'})$
- $instance(TradeWebsite, \text{'marktplaats.nl'})$
- $instance(AdvertisementTitle, \text{'Barbie doll'})$
- $instance(AdvertisementNumber, \text{'m0001'})$
- $instance(TransactionDate, \text{'09-10-2017'})$
- $instance(PurchaseMoney, \text{'€33,95'})$
- $instance(PaymentMethod, \text{'Bank transfer (IBAN)'})$
- $instance(BankAccountCP, \text{'NL00BANK1234'})$
- $instance(AccountHolderCP, \text{'Lukas Goedeman'})$
- $instance(BankAccountA, \text{'NL00CASH9999'})$
- $instance(ProductType, \text{'Barbie doll'})$

Intake 4:

- $OrderedByA$
- $OfferedByCP$
- $PaymentByA$
- $\neg ContactAfter$
- $TrackAndTrace$
- $instance(NameA, \text{'Applicant 3'})$
- $instance(PhoneNumberA, \text{'020-3333'})$
- $instance(MobilePhoneA, \text{'06-3333'})$
- $instance(UsernameCP, \text{'Toys4ever'})$
- $instance(CountryCP, \text{'Netherlands'})$
- $instance(TradeWebsite, \text{'marktplaats.nl'})$
- $instance(AdvertisementNumber, \text{'m0002'})$
- $instance(TransactionDate, \text{'06-10-2017'})$
- $instance(PurchaseMoney, \text{'€126,95'})$
- $instance(PaymentMethod, \text{'Bank transfer (IBAN)'})$
- $instance(BankAccountCP, \text{'NL00BANK1234'})$
- $instance(AccountHolderCP, \text{'Lukas Goedeman'})$
- $instance(TransactionTime, \text{'17:45'})$
- $instance(BankAccountA, \text{'NL00MUNT6666'})$
- $instance(ProductType, \text{'Dikkie Dik book series (11-part)'})$

Figure 21: Observations based on Intake 3 & 4

7.2.5 Matching

In section 5.7, we introduced the rule set $\mathcal{R}_m$ for matching police reports. We use function q to add $\{R_1, R_2, R_3, R_4\}$ to the topic language. Both r_m1 and r_m2 could be used to match the four reports, but only one of them is required. Say that the intake agent uses r_m1, then it could form the following argument:

A_1: $observed(instance(BankAccountCP, \text{'NL00BANK1234'}), R_1)$

A_2: $observed(instance(BankAccountCP, \text{'NL00BANK1234'}), R_2)$

A_3: $observed(instance(BankAccountCP, \text{'NL00BANK1234'}), R_3)$

A_4: $observed(instance(BankAccountCP, \text{'NL00BANK1234'}), R_4)$

A_5: $A_1, A_2 \Rightarrow match(R_1, R_2)$

A_6: $A_2, A_3 \Rightarrow match(R_2, R_3)$

A_7: $A_3, A_4 \Rightarrow match(R_3, R_4)$

The following dialogue ensues:

1. $P_{int} : begin()$
    2. $P_{int} : argue(R_1), argue(R_2), argue(R_3), argue(R_4), claim(match(R_1, R_2)), claim(match(R_2, R_3)), claim(match(R_3, R_4))$
        3. $P_{ana} : concede(R_1), concede(R_2), concede(R_3), concede(R_4), concede(match(R_1, R_2)), concede(match(R_2, R_3)), concede(match(R_3, R_4))$
4. $P_{int} : finish()$

7.3 Case File Dialogue

Having received the information given by the intake agent, the initial belief base of the analysis agent is $B^0_{ana} = \{R_1, Prop(R_1), R_2, Prop(R_2), R_3, Prop(R_3), R_4, Prop(R_4), match(R_1, R_2), match(R_2, R_3), match(R_3, R_4)\}$. We will assume there are two relevant external contacts for this particular case, namely the Marktplaats agent $P_{ext.m}$ (with its corresponding inquiring agent $P_{inq.m}$) and the Bank agent $P_{ext.b}$ (with its corresponding inquiring agent $P_{inq.b}$). Given that the dialogue system is $\mathcal{D}^{cd}$

with the set of participants $\mathcal{P}^{cd} = \{P_{ana}, P_{inq.m}, P_{inq.b}, P_{ext.m}, P_{ext.b}\}$, and that the argumentation system is AS^{cd}, we construct a possible case file dialogue. The dialogue and the resulting argument can be found in Appendix C. This case file dialogue is completely fabricated in order to demonstrate the workings of the established protocol and rules and is not based on an actual investigation. The results do not reflect the findings of the police, and all acquired data is deliberately left vague.

In the dialogue, the analysis agent demands an investigation about two topics, namely the Marktplaats account of the suspect, and their bank account. The Marktplaats agent can provide an e-mail address, a bank account and an IP address, while the Bank agent can provide a name, an address, a residence, a different e-mail address, a phone number, and naturally a bank account. The fact that the name connected to the bank account is not the name that the counterparty has used in the advertisements, is grounds to suspect that a fake name is used. The final argument acknowledges this, along with an identity description of the suspect that is as comprehensive as possible and a conflict description based on the reports (in this case, the compositional argument of $\{R_1, R_2, R_3, R_4\}$).

8 Conclusion and discussion

In this thesis, we have constructed an overarching dialogue framework for the intake and case file process of police reports, combined with argumentation systems specifically designed for cases of fraud. Given a set of fraud reports and the means to textually interpret them, the agents in the framework should be able to adequately determine whether each of them presumably concerns fraud, to retrieve missing information through intake dialogues, to match the reports that concern the same conflict, to investigate the conflicts through case file dialogues, and to produce concluding arguments containing the conflict description and suspect identification for each case.

We can answer our first two research questions, 'Can we construct a dialogue system for intake dialogues through which the validity of fraud claims is established and missing information can be requested?' and 'Can we construct a dialogue system for case file dialogue, that combines information from police reports and external sources in order to formulate a coherent argument?', affirmingly. The framework consists of two main dialogue systems: The intake dialogue system $\mathcal{D}^{id}$ and the case file dialogue system $\mathcal{D}^{cd}$. The protocol of $\mathcal{D}^{id}$ contains a validation exchange for establishing whether the conflict presumably concerns a crime (specifically fraud under the used rule set $\mathcal{R}^{id}$), and a specification exchange used to request missing data from the applicant. Through $\mathcal{D}^{cd}$, information established during the intake process can be investigated in order to establish the identity of the suspect and build a case against them.

We can favourably answer the third research question, 'Can we combine these two overarching dialogue protocols into a single formal framework?', as well. The two systems differ in many ways, but are interconnected due to the fact that any case file dialogue is directly dependent on the resulting arguments of one or more intake dialogues. The intake agent and analysis agent are in direct contact with each other, as the intake agent submits all successful reports to the analysis agent, and the analysis agent can give feedback to the intake agent. Furthermore, all argumentation rules that apply to $\mathcal{D}^{id}$ are carried over to $\mathcal{D}^{cd}$.

To answer the the final research question, 'Can we structure the resulting arguments in a way that they can be transcribed into natural language?', we need to look at the complete rule set $\mathcal{R}^{cd}$. Given that a report is accepted, the concluding arguments in both dialogue types are designed to represent a description of the conflict, which includes an explanation on how this presumably involves a violation of law, as well as to provide the relevant details of the involved parties. Transcribing the arguments in natural language is relatively straightforward, since the argumentation rules for fraud are consciously set up in a way that an explainable conflict narrative can be built based on the observations of the applicants. In contrast, the argumentation rules for the investigations and inquiries do not shape such a clear narrative, but instead sum up the known attributes of the person of interest.

Both dialogue systems are inspired by different properties of established dialogue types, but have significantly more complex dialogue protocols than any of the systems on which they are based. Where the protocol of an information-seeking dialogue, persuasion dialogue or inquiry dialogue can be summed up with a few sentences (as shown in section 3), the protocols of the intake dialogue and the case file dialogue each require three levels of sub-protocols, each sub-protocol describes a relatively large number of possible situations and each situation has very case-specific responses and follow-ups. Although this allows the systems to handle very specific scenarios in order to arrive at a desirable result, this is at the expense of the dialectic freedom the more basic dialogue systems possess. Furthermore, a lot of clarity is lost in the process, as the move-set cannot be described in generalist terms such as attacks and surrenders. In a way, a large portion of the agents' strategies are already embedded in the protocol. For future research, a more straightforward variant of the current dialogue system could be designed, such that there would be fewer restrictions on possible moves, while the strategy descriptions for each of the agents would be more extensive and diverse.

Concerning strategies, we have consistently assumed exhaustive strategies for both the intake agent and the case file agent. This could result in long dialogues and extensive arguments (see the case study for an example of this). For real-life applications, this might cause participants to prematurely quit the dialogue or stop responding. There is currently research in progress about strategy selection through reinforcement learning, based on existing police reports.[34] The results of this research could prove to be useful for determining more effective strategies within the current framework.

What might be a useful addition to the case file dialogue system, but is beyond the scope of

[34] Floris Bex, personal communication.

this thesis, is an additional analysis of the investigation results. For instance, if multiple people are involved in the scam, how can we assign the pieces of information to the relevant party? Should some external contacts be preferred over others, and if so, what does this entail in practice? How can we make a clear overview of the financial picture in each case? Hopefully, further research can use the dialogue framework of this thesis as a starting point for automation of more elaborate case files.

References

[1] J.F. Bergers. Het verbeteren van online aangifte voor online handelsfraude door het maken van een argumentatietheorie (Niet geanonimiseerd). Utrecht University, 2018.

[2] J. Bernklau and J. van der Putte. Overzicht jurisprudentie internetoplichting LMIO. Openbaar Ministerie, 2015.

[3] F.J. Bex, J. Peters, and B.J.G. Testerink. AI for Online Criminal Complaints: From Natural Dialogues to Structured Scenarios. In *Artificial Intelligence for Justice Workshop (ECAI 2016)*, page 22, 2016.

[4] F.J. Bex and H. Prakken. Investigating stories in a formal dialogue game. *Computational Models of Argument. Proceedings of COMMA 2008. Frontiers in Artificial Intelligence and Applications*, 172:73, 2008.

[5] E. Black and A. Hunter. An inquiry dialogue system. *Autonomous Agents and Multi-Agent Systems*, 19(2):173–209, 2009.

[6] W. Kos, M.P. Schraagen, M.J.S. Brinkhuis, and F.J. Bex. Classification in a Skewed Online Trade Fraud Complaint Corpus. In *Preproceedings of the 29th Benelux Conference on Artificial Intelligence November 8–9, 2017 in Groningen, The Netherlands*, pages 172–183, 2017.

[7] R. Latukolan and J. van Ginkel. Het in kaart brengen van het proces bij initiatie van een melding tot het opstellen van een dossier. *Unpublished internal document*, 2016.

[8] S. Modgil and H. Prakken. The ASPIC+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.

[9] S. Parsons, P. McBurney, and M. Wooldridge. The Mechanics of Some Formal Inter-agent Dialogues. In *Workshop on Agent Communication Languages*, volume 2922, pages 329–348. Springer, 2003.

[10] S. Parsons, M. Wooldridge, and L. Amgoud. On the Outcomes of Formal Inter-agent Dialogues. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*.

[11] S. Parsons, M. Wooldridge, and L. Amgoud. An Analysis of Formal Inter-agent Dialogues. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 394–401. ACM, 2002.

[12] Politie. Aangifte of melding doen: (Internet)oplichting. https://www.politie.nl/aangifte-of-melding-doen#link-internetoplichting. Accessed: 2018-02-09.

[13] H. Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, 15(6):1009–1040, 2005.

[14] H. Prakken. Formal systems for persuasion dialogue. *The knowledge engineering review*, 21(2):163–188, 2006.

[15] C. Reed. Dialogue Frames in Agent Communication. In *Multi Agent Systems, 1998. Proceedings. International Conference*, pages 246–253. IEEE, 1998.

[16] M.P. Schraagen, M.J.S. Brinkhuis, and F.J. Bex. Evaluation of named entity recognition in dutch online criminal complaints. *Computational Linguistics in The Netherlands journal*, 7:3–16, 2017.

[17] I.L.N. van 't Hul. Improving online trade fraud complaint classification by applying machine learning techniques. Utrecht University, 2018.

[18] D.N. Walton and E.C.W. Krabbe. *Commitment in dialogue: Basic concepts of interpersonal reasoning*. SUNY press, 1995.

Appendices

A Full argument A_{intake} for 5.6

A_1 : $OfferedByCP$

A_2 : $\neg DeliveredByCP$

A_3 : $Waited5Days$

A_4 : $A_2, A_3 \Rightarrow \neg SentByCP$

A_5 : $ProductPictured$

A_6 : $A_5 \Rightarrow LegitimateSalesIndicator$

A_7 : $A_1, A_4, A_6 \Rightarrow DeceitfulSalesmanCP$

A_8 : $A_7 \Rightarrow FraudScheme$

A_9 : $OrderedByA$

A_{10} : $A_1, A_9 \Rightarrow MutualAgreement$

A_{11} : $ContactOverEmail$

A_{12} : $A_{11} \Rightarrow ContactPrior$

A_{13} : $\neg RespondMessageCP$

A_{14} : $A_{13} \Rightarrow \neg ContactAfter$

A_{15} : $A_{12}, A_{14} \Rightarrow BreakingContact$

A_{16} : $A_8, A_{10}, A_{15} \Rightarrow IntendedFraudulentBenefit$

A_{17} : $PaymentByA$

A_{18} : $A_9, A_{17} \Rightarrow ObtainingGoods$

A_{19} : $A_{16}, A_{18} \Rightarrow pFraud$

A_{pCrime} : $A_{19} \rightarrow pCrime$

A_{20} : $instance(FirstnameCP, \text{‘Johannes’})$

A_{21} : $A_{20} \rightarrow known(FirstnameCP)$

A_{22} : $A_{21} \rightarrow check(FirstnameCP)$

A_{23} : $instance(SurnamePrefixCP, \text{‘de’})$

A_{24} : $A_{23} \rightarrow known(SurnamePrefixCP)$

A_{25} : $A_{24} \rightarrow check(SurnamePrefixCP)$

A_{26} : $instance(SurnameCP, \text{‘Silentio’})$

A_{27} : $A_{26} \rightarrow known(SurnameCP)$

A_{28} : $A_{27} \rightarrow check(SurnameCP)$

A_{29} : $instance(EmailAddressCP, \text{‘fear@trembling.nl’})$

A_{30} : $A_{29} \rightarrow known(EmailAddressCP)$

A_{31} : $A_{30} \rightarrow check(EmailAddressCP)$

A_{32} : $\neg known(PhoneNumberCP)$

A_{33} : $A_{32} \rightarrow check(PhoneNumberCP)$

A_{34} : $A_{22}, A_{25}, A_{28}, A_{31}, A_{33} \rightarrow InfoCPComplete$

A_{35} : $instance(BankAccountCP, \text{‘NL99BANK0112358132 1’})$

A_{36} : $A_{35} \rightarrow known(BankAccountCP)$

A_{37} : $A_{36} \rightarrow check(BankAccountCP)$

A_{38} : $instance(TradeWebsite, \text{‘Marktplaats’})$

A_{39} : $A_{38} \rightarrow known(TradeWebsite)$

A_{40} : $A_{39} \rightarrow check(TradeWebsite)$

A_{41} : $instance(PurchaseMoney, \text{`€270'})$

A_{42} : $A_{41} \rightarrow known(PurchaseMoney)$

A_{43} : $A_{42} \rightarrow check(PurchaseMoney)$

A_{44} : $instance(ProductType, \text{`NintendoSwitch'})$

A_{45} : $A_{44} \rightarrow known(ProductType)$

A_{46} : $A_{45} \rightarrow check(ProductType)$

A_{47} : $A_{37}, A_{40}, A_{43}, A_{46} \rightarrow InfoConflictComplete$

A_{Intake} : $A_{34}, A_{47} \rightarrow Intake$

$A_{Complete}$: $A_{pCrime}, A_{Intake} \rightarrow A_{Complete}$

B Intake dialogues

B.1 Intake dialogue 1

1. P_{int}: $begin(Intake)$

2. P_{int}: $begin(pCrime)$

3. P_{app}: $claim(OrderedByA)$

4. P_{int}: $concede(OrderedByA)$

5. P_{app}: $claim(OfferedByCP)$

6. P_{int}: $concede(OfferedByCP)$

7. P_{app}: $claim(PaymentByA)$

8. P_{int}: $concede(PaymentByA)$

9. P_{app}: $claim(\neg DeliveredByCP)$

10. P_{int}: $concede(\neg DeliveredByCP)$

11. P_{app}: $claim(\neg RespondMessageCP)$

12. P_{int}: $concede(\neg RespondMessageCP)$

13. P_{app}: $claim(TrackAndTrace)$

14. P_{int}: $concede(TrackAndTrace)$

15. P_{int}: $ask(Waited5Days)$

16. P_{app}: $claim(Waited5Days)$

17. P_{int}: $concede(Waited5Days)$

18. P_{int}: $ask(ContactPrior)$

19. P_{app}: $claim(ContactPrior)$

20. P_{int}: $challenge(ContactPrior)$

21. P_{app}: $argue(ContactThroughTradeWebsite \Rightarrow ContactPrior)$

22. P_{int}: $concede(ContactThroughTradeWebsite \Rightarrow ContactPrior)$

23. P_{int}: $argue(((((((OfferedByCP, (\neg DeliveredByCP, Waited5Days \Rightarrow \neg SentByCP), (TrackAndTrace \Rightarrow LegitimateSalesIndicator)) \Rightarrow DeceitfulSalesmanCP) \Rightarrow FraudScheme), (OfferedByCP, OrderedByA \Rightarrow MutualAgreement), ((ContactThroughTradewebsite \Rightarrow ContactPrior), (\neg RespondMessageCP \Rightarrow \neg ContactAfter) \Rightarrow BreakingContact) \Rightarrow IntendedFraudulentBenefit), (OrderedByA, PaymentByA \Rightarrow ObtainingGoods)) \Rightarrow pFraud) \rightarrow pCrime)$

24. P_{int}: $finish(pCrime)$

25. P_{int}: $begin(Complete)$

26. P_{app}: $claim(instance(NameA, \text{`Applicant 1'}))$

27. P_{int}: $concede(instance(NameA, \text{`Applicant 1'}))$

28. P_{app}: $claim(instance(PhoneNumberA, \text{`06-1111'}))$

29. P_{int}: $concede(instance(PhoneNumberA, \text{`06-1111*'}))$

30. P_{app}: $claim(instance(MobilePhoneA, \text{`06-1111'}))$

31. P_{int}: $concede(instance(MobilePhoneA, \text{`06-1111'}))$

32. P_{app}: $claim(instance(NameCP, \text{`Dennis de Boer'}))$

33. P_{int}: $concede(instance(NameCP, \text{'Dennis de Boer'}))$

34. P_{app}: $claim(instance(UsernameCP, \text{'Toys4ever'}))$

35. P_{int}: $concede(instance(UsernameCP, \text{'Toys4ever'}))$

36. P_{app}: $claim(instance(ResidenceCP, \text{'Volendam'}))$

37. P_{int}: $concede(instance(ResidenceCP, \text{'Volendam'}))$

38. P_{app}: $claim(instance(CountryCP, \text{'Netherlands'}))$

39. P_{int}: $concede(instance(CountryCP, \text{'Netherlands'}))$

40. P_{app}: $claim(instance(TradeWebsite, \text{'marktplaats.nl'}))$

41. P_{int}: $concede(instance(TradeWebsite, \text{'marktplaats.nl'}))$

42. P_{app}: $claim(instance(AdvertisementTitle, \text{'Barbie doll'}))$

43. P_{int}: $concede(instance(AdvertisementTitle, \text{'Barbie doll'}))$

44. P_{app}: $claim(instance(AdvertisementNumber, \text{'m0001'}))$

45. P_{int}: $concede(instance(AdvertisementNumber, \text{'m0001'}))$

46. P_{app}: $claim(instance(TransactionDate, \text{'08-10-2017'}))$

47. P_{int}: $concede(instance(TransactionDate, \text{'08-10-2017'}))$

48. P_{app}: $claim(instance(PurchaseMoney, \text{'€30'}))$

49. P_{int}: $concede(instance(PurchaseMoney, \text{'€30'}))$

50. P_{app}: $claim(instance(PaymentMethod, \text{'Bank transfer (IBAN)'}))$

51. P_{int}: $concede(instance(PaymentMethod, \text{'Bank transfer (IBAN)'}))$

52. P_{app}: $claim(instance(BankAccountCP, \text{'NL00BANK1234'}))$

53. P_{int}: $concede(instance(BankAccountCP, \text{'NL00BANK1234'}))$

54. P_{app}: $claim(instance(AccountHolderCP, \text{'Dennis de Boer'}))$

55. P_{int}: $concede(instance(AccountHolderCP, \text{'Dennis de Boer'}))$

56. P_{app}: $claim(instance(AccountHolderCP, \text{'Lukas Goedeman'}))$

57. P_{int}: $argue(instance(AccountHolderCP, \text{'Dennis de Boer'}) \Rightarrow \neg instance(AccountHolderCP, \text{'Lukas Goedeman'}))$

58. P_{app}: $retract(instance(AccountHolderCP, \text{'Lukas Goedeman'}))$

59. P_{app}: $claim(instance(TransactionTime, \text{'11:22'}))$

60. P_{int}: $concede(instance(TransactionTime, \text{'11:22'}))$

61. P_{app}: $claim(instance(BankAccountA, \text{'NL00GELD7777'}))$

62. P_{int}: $concede(instance(BankAccountA, \text{'NL00GELD7777'}))$

63. P_{app}: $claim(instance(ProductType, \text{'Barbie doll'}))$

64. P_{int}: $concede(instance(ProductType, \text{'Barbie doll'}))$

65. P_{int}: $argue(check(PhoneNumberA), check(NameCP), check(UsernameCP), check(TradeWebsite), check(TransActionDate), check(PurchaseMoney), check(PaymentMethod), check(BankAccountCP), check(AccountHolderCP), check(TransactionTime), check(BankAccountA), check(ProductType) \rightarrow Complete)$

66. P_{int}: $finish(Complete)$

67. P_{int}: $argue(pCrime, Complete \rightarrow Intake)$

68. P_{int}: $finish(Intake)$

B.2 Intake dialogue 2

1. P_{int}: $begin(Intake)$

2. P_{int}: $begin(pCrime)$

3. P_{app}: $claim(OrderedByA)$

4. P_{int}: $concede(OrderedByA)$

5. P_{app}: $claim(OfferedByCP)$

6. P_{int}: $concede(OfferedByCP)$

7. P_{app}: $claim(PaymentByA)$

8. P_{int}: $concede(PaymentByA)$

9. P_{app}: $claim(\neg DeliveredByCP)$

10. P_{int}: $concede(\neg DeliveredByCP)$

11. P_{app}: $claim(\neg RespondMessageCP)$

12. P_{int}: $concede(\neg RespondMessageCP)$

13. P_{app}: $claim(TrackAndTrace)$

14. P_{int}: $concede(TrackAndTrace)$

15. P_{int}: $ask(Waited5Days)$

16. P_{app}: $claim(Waited5Days)$

17. P_{int}: $concede(Waited5Days)$

18. P_{int}: $ask(ContactPrior)$

19. P_{app}: $claim(ContactPrior)$

20. P_{int}: $challenge(ContactPrior)$

21. P_{app}: $argue(ContactThroughTradeWebsite \Rightarrow ContactPrior)$

22. P_{int}: $concede(ContactThroughTradeWebsite \Rightarrow ContactPrior)$

23. P_{int}: $argue(((((((OfferedByCP, (\neg DeliveredByCP, Waited5Days \Rightarrow \neg SentByCP), (TrackAndTrace \Rightarrow LegitimateSalesIndicator)) \Rightarrow DeceitfulSalesmanCP) \Rightarrow FraudScheme), (OfferedByCP, OrderedByA \Rightarrow MutualAgreement), ((ContactThroughTradewebsite \Rightarrow ContactPrior), (\neg RespondMessageCP \Rightarrow \neg ContactAfter) \Rightarrow BreakingContact) \Rightarrow IntendedFraudulentBenefit), (OrderedByA, PaymentByA \Rightarrow ObtainingGoods)) \Rightarrow pFraud) \rightarrow pCrime)$

24. P_{int}: $finish(pCrime)$

25. P_{int}: $begin(Complete)$

26. P_{app}: $claim(instance(NameA, \text{'Applicant 1'}))$

27. P_{int}: $concede(instance(NameA, \text{'Applicant 1'}))$

28. P_{app}: $claim(instance(PhoneNumberA, \text{'06-1111'}))$

29. P_{int}: $concede(instance(PhoneNumberA, \text{'06-1111'}))$

30. P_{app}: $claim(instance(MobilePhoneA, \text{'06-1111'}))$

31. P_{int}: $concede(instance(MobilePhoneA, \text{'06-1111'}))$

32. P_{app}: $claim(instance(NameCP, \text{'Dennis de Boer'}))$

33. P_{int}: $concede(instance(NameCP, \text{'Dennis de Boer'}))$

34. P_{app}: $claim(instance(UsernameCP, \text{'Toys4ever'}))$

35. P_{int}: $concede(instance(UsernameCP, \text{'Toys4ever'}))$

36. P_{app}: $claim(instance(ResidenceCP, \text{'Volendam'}))$

37. P_{int}: $concede(instance(ResidenceCP, \text{'Volendam'}))$

38. P_{app}: $claim(instance(CountryCP, \text{'Netherlands'}))$

39. P_{int}: $concede(instance(CountryCP, \text{'Netherlands'}))$

40. P_{app}: $claim(instance(TradeWebsite, \text{'marktplaats.nl'}))$

41. P_{int}: $concede(instance(TradeWebsite, \text{'marktplaats.nl'}))$

42. P_{app}: $claim(instance(AdvertisementTitle, \text{'Dikkie Dik books'}))$

43. P_{int}: $concede(instance(AdvertisementTitle, \text{Dikkie Dik books'}))$

44. P_{app}: $claim(instance(AdvertisementNumber, \text{'m0002'}))$

45. P_{int}: $concede(instance(AdvertisementNumber, \text{'m0002'}))$

46. P_{app}: $claim(instance(TransactionDate, \text{'08-10-2017'}))$

47. P_{int}: $concede(instance(TransactionDate, \text{'08-10-2017'}))$

48. P_{app}: $claim(instance(PurchaseMoney, \text{'€41,95'}))$

49. P_{int}: $concede(instance(PurchaseMoney, \text{'€41,95'}))$

50. P_{app}: $claim(instance(PaymentMethod, \text{'Bank transfer (IBAN)'}))$

51. P_{int}: $concede(instance(PaymentMethod, \text{'Bank transfer (IBAN)'}))$

52. P_{app}: $claim(instance(BankAccountCP, \text{'NL00BANK1234'}))$

53. P_{int}: $concede(instance(BankAccountCP, \text{'NL00BANK1234'}))$

54. P_{app}: $argue(instance(AccountHolderCP, \text{'Dennis de Boer'}) \Rightarrow \neg instance(AccountHolderCP, \text{'Lukas Goedeman'}))$

55. P_{int}: $concede(instance(AccountHolderCP, \text{'Dennis de Boer'}) \Rightarrow \neg instance(AccountHolderCP, \text{'Lukas Goedeman'}))$

56. P_{app}: $claim(instance(TransactionTime, \text{'11:22'}))$

57. P_{int}: $concede(instance(TransactionTime, \text{'11:22'}))$

58. P_{app}: $claim(instance(BankAccountA, \text{'NL00GELD7777'}))$

59. P_{int}: $concede(instance(BankAccountA, \text{'NL00GELD7777'}))$

60. P_{app}: $claim(instance(ProductType, \text{'Dikkie Dik book series (8-part)'}))$

61. P_{int}: $concede(instance(ProductType, \text{'Dikkie Dik book series (8-part)'}))$

62. P_{int}: $argue(check(PhoneNumberA), check(NameCP), check(UsernameCP), check(TradeWebsite), check(TransActionDate), check(PurchaseMoney), check(PaymentMethod), check(BankAccountCP), check(AccountHolderCP), check(TransactionTime), check(BankAccountA), check(ProductType) \rightarrow Complete)$

63. P_{int}: $finish(Complete)$

64. P_{int}: $argue(pCrime, Complete \rightarrow Intake)$

65. P_{int}: $finish(Intake)$

B.3 Intake dialogue 3

1. P_{int}: $begin(Intake)$

2. P_{int}: $begin(pCrime)$

3. P_{app}: $claim(OrderedByA)$

4. P_{int}: $concede(OrderedByA)$

5. P_{app}: $claim(OfferedByCP)$

6. P_{int}: $concede(OfferedByCP)$

7. P_{app}: $claim(PaymentByA)$

8. P_{int}: $concede(PaymentByA)$

9. P_{app}: $claim(\neg ContactAfter)$

10. P_{int}: $challenge(\neg ContactAfter)$

11. P_{int}: $argue(\neg RespondMessageCP \Rightarrow \neg ContactAfter)$

12. P_{int}: $ask(DeliveredByCP)$

13. P_{app}: $claim(\neg DeliveredByCP)$

14. P_{int}: $concede(\neg DeliveredByCP)$

15. P_{int}: $ask(Waited5Days)$

16. P_{app}: $claim(Waited5Days)$

17. P_{int}: $concede(Waited5Days)$

18. P_{int}: $ask(ContactPrior)$

19. P_{app}: $claim(ContactPrior)$

20. P_{int}: $challenge(ContactPrior)$

21. P_{app}: $argue(ContactThroughTradeWebsite \Rightarrow ContactPrior)$

22. P_{int}: $concede(ContactThroughTradeWebsite \Rightarrow ContactPrior)$

23. P_{int}: $ask(LegitimateSaleIndicator)$

24. P_{app}: $argue(ProductPictured \Rightarrow LegitimateSaleIndicator)$

25. P_{int}: $concede(ProductPictured \Rightarrow LegitimateSaleIndicator)$

26. P_{int}: $argue(((((((OfferedByCP, (\neg DeliveredByCP, Waited5Days \Rightarrow \neg SentByCP), (ProductPictured \Rightarrow LegitimateSalesIndicator)) \Rightarrow DeceitfulSalesmanCP) \Rightarrow FraudScheme), (OfferedByCP, OrderedByA \Rightarrow MutualAgreement), ((ContactThroughTradewebsite \Rightarrow ContactPrior), (\neg RespondMessageCP \Rightarrow \neg ContactAfter) \Rightarrow BreakingContact) \Rightarrow IntendedFraudulentBenefit), (OrderedByA, PaymentByA \Rightarrow ObtainingGoods)) \Rightarrow pFraud) \rightarrow pCrime)$

27. P_{int}: $finish(pCrime)$

28. P_{int}: $begin(Complete)$

29. P_{app}: $claim(instance(NameA, \text{'Applicant 2'}))$

30. P_{int}: $concede(instance(NameA, \text{'Applicant 2'}))$

31. P_{app}: $claim(instance(PhoneNumberA, \text{'06-2222'}))$

32. P_{int}: $concede(instance(PhoneNumberA, \text{'06-2222'}))$

33. P_{app}: $claim(instance(MobilePhoneA, \text{'06-2222'}))$

34. P_{int}: $concede(instance(MobilePhoneA, \text{'06-2222'}))$

35. P_{app}: $claim(instance(NameCP, \text{'Lukas Goedeman'}))$

36. P_{int}: $concede(instance(NameCP, \text{'Lukas Goedeman'}))$

37. P_{app}: $claim(instance(UsernameCP, \text{'Toys4ever'}))$

38. P_{int}: $concede(instance(UsernameCP, \text{'Toys4ever'}))$

39. P_{app}: $claim(instance(CountryCP, \text{'Netherlands'}))$

40. P_{int}: $concede(instance(CountryCP, \text{'Netherlands'}))$

41. P_{app}: $claim(instance(TradeWebsite, \text{'marktplaats.nl'}))$

42. P_{int}: $concede(instance(TradeWebsite, \text{'marktplaats.nl'}))$

43. P_{app}: $claim(instance(AdvertisementTitle, \text{'Barbie doll'}))$

44. P_{int}: $concede(instance(AdvertisementTitle, \text{'Barbie doll'}))$

45. P_{app}: $claim(instance(AdvertisementNumber, \text{'m0001'}))$

46. P_{int}: $concede(instance(AdvertisementNumber, \text{'m0001'}))$

47. P_{app}: $claim(instance(TransactionDate, \text{'09-10-2017'}))$

48. P_{int}: $concede(instance(TransactionDate, \text{'09-10-2017'}))$

49. P_{app}: $claim(instance(PurchaseMoney, \text{'€33.95'}))$

50. P_{int}: $concede(instance(PurchaseMoney, \text{'€33.95'}))$

51. P_{app}: $claim(instance(PaymentMethod, \text{'iDEAL (IBAN)'}))$

52. P_{int}: $concede(instance(PaymentMethod, \text{'iDEAL (IBAN)'}))$

53. P_{app}: $claim(instance(BankAccountCP, \text{'NL00BANK1234'}))$

54. P_{int}: $concede(instance(BankAccountCP, \text{'NL00BANK1234'}))$

55. P_{app}: $claim(instance(AccountHolderCP, \text{'Lukas Goedeman'}))$

56. P_{int}: $concede(instance(AccountHolderCP, \text{'Lukas Goedeman'}))$

57. P_{app}: $claim(instance(BankAccountA, \text{'NL00CASH9999'}))$

58. P_{int}: $concede(instance(BankAccountA, \text{'NL00CASH9999'}))$

59. P_{app}: $claim(instance(ProductType, \text{'Barbie doll'}))$

60. P_{int}: $concede(instance(ProductType, \text{'Barbie doll'}))$

61. P_{int}: $ask(known(TransactionTime))$

62. P_{int}: $claim(\neg known(TransactionTime))$

63. P_{int}: $concede(\neg known(TransactionTime))$

64. P_{int}: $argue(check(PhoneNumberA), check(NameCP), check(UsernameCP), check(TradeWebsite), check(TransActionDate), check(PurchaseMoney), check(PaymentMethod), check(BankAccountCP), check(AccountHolderCP), check(TransactionTime), check(BankAccountA), check(ProductType) \rightarrow Complete)$

65. P_{int}: $finish(Complete)$

66. P_{int}: $argue(pCrime, Complete \rightarrow Intake)$

67. P_{int}: $finish(Intake)$

B.4 Intake dialogue 4

1. P_{int}: $begin(Intake)$

2. P_{int}: $begin(pCrime)$

3. P_{app}: $claim(OrderedByA)$

4. P_{int}: $concede(OrderedByA)$

5. P_{app}: $claim(OfferedByCP)$

6. P_{int}: $concede(OfferedByCP)$

7. P_{app}: $claim(PaymentByA)$

8. P_{int}: $concede(PaymentByA)$

9. P_{app}: $claim(\neg ContactAfter)$

10. P_{int}: $challenge(\neg ContactAfter)$

11. P_{int}: $argue(\neg RespondMessageCP \Rightarrow \neg ContactAfter)$

12. P_{app}: $claim(TrackAndTrace)$

13. P_{int}: $concede(TrackAndTrace)$

14. P_{int}: $ask(DeliveredByCP)$

15. P_{app}: $claim(\neg DeliveredByCP)$

16. P_{int}: $concede(\neg DeliveredByCP)$

17. P_{int}: $ask(Waited5Days)$

18. P_{app}: $claim(Waited5Days)$

19. P_{int}: $concede(Waited5Days)$

20. P_{int}: $ask(ContactPrior)$

21. P_{app}: $claim(ContactPrior)$

22. P_{int}: $challenge(ContactPrior)$

23. P_{app}: $argue(ContactThroughTradeWebsite \Rightarrow ContactPrior)$

24. P_{int}: $concede(ContactThroughTradeWebsite \Rightarrow ContactPrior)$

25. P_{int}: $argue(((((((OfferedByCP, (\neg DeliveredByCP, Waited5Days \Rightarrow \neg SentByCP), (TrackAndTrace \Rightarrow LegitimateSalesIndicator)) \Rightarrow DeceitfulSalesmanCP) \Rightarrow FraudScheme), (OfferedByCP, OrderedByA \Rightarrow MutualAgreement), ((ContactThroughTradewebsite \Rightarrow ContactPrior), (\neg RespondMessageCP \Rightarrow \neg ContactAfter) \Rightarrow BreakingContact) \Rightarrow IntendedFraudulentBenefit), (OrderedByA, PaymentByA \Rightarrow ObtainingGoods)) \Rightarrow pFraud) \rightarrow pCrime)$

26. P_{int}: $finish(pCrime)$

27. P_{int}: $begin(Complete)$

28. P_{app}: $claim(instance(NameA, \text{'Applicant 3'}))$

29. P_{int}: $concede(instance(NameA, \text{'Applicant 3'}))$

30. P_{app}: $claim(instance(PhoneNumberA, \text{'020-3333'}))$

31. P_{int}: $concede(instance(PhoneNumberA, \text{'020-3333'}))$

32. P_{app}: $claim(instance(MobilePhoneA, \text{'06-3333'}))$

33. P_{int}: $concede(instance(MobilePhoneA, \text{'06-3333'}))$

34. P_{app}: $claim(instance(UsernameCP, \text{'Toys4ever'}))$

35. P_{int}: $concede(instance(UsernameCP, \text{'Toys4ever'}))$

36. P_{app}: $claim(instance(CountryCP, \text{'Netherlands'}))$

37. P_{int}: $concede(instance(CountryCP, \text{'Netherlands'}))$

38. P_{app}: $claim(instance(TradeWebsite, \text{'marktplaats.nl'}))$

39. P_{int}: $concede(instance(TradeWebsite, \text{'marktplaats.nl'}))$

40. P_{app}: $claim(instance(AdvertisementNumber, \text{'m0002'}))$

41. P_{int}: $concede(instance(AdvertisementNumber, \text{'m0002'}))$

42. P_{app}: $claim(instance(TransactionDate, \text{'06-10-2017'}))$

43. P_{int}: $concede(instance(TransactionDate, \text{'06-10-2017'}))$

44. P_{app}: $claim(instance(PurchaseMoney, \text{'€126,95'}))$

45. P_{int}: $concede(instance(PurchaseMoney, \text{'€126,95'}))$

46. P_{app}: $claim(instance(PaymentMethod, \text{'Bank transfer (IBAN)'}))$

47. P_{int}: $concede(instance(PaymentMethod, \text{'Bank transfer (IBAN)'}))$

48. P_{app}: $claim(instance(BankAccountCP, \text{'NL00BANK1234'}))$

49. P_{int}: $concede(instance(BankAccountCP, \text{'NL00BANK1234'}))$

50. P_{app}: $claim(instance(AccountHolderCP, \text{'Lukas Goedeman'}))$

51. P_{int}: $concede(instance(AccountHolderCP, \text{'Lukas Goedeman'}))$

52. P_{app}: $claim(instance(TransactionTime, \text{'17:45'}))$

53. P_{int}: $concede(instance(TransactionTime, \text{'17:45'}))$

54. P_{app}: $claim(instance(BankAccountA, \text{'NL00MUNT6666'}))$

55. P_{int}: $concede(instance(BankAccountA, \text{'NL00MUNT6666'}))$

56. P_{app}: $claim(instance(ProductType, \text{'Dikkie Dik book series (11-part)'}))$

57. P_{int}: $concede(instance(ProductType, \text{'Dikkie Dik book series (11-part)'}))$

58. P_{int}: $ask(known(nameCP))$

59. P_{app}: $claim(instance(nameCP, \text{'Lukas Goedeman'}))$

60. P_{int}: $concede(instance(nameCP, \text{'Lukas Goedeman'}))$

61. P_{int}: $argue(check(PhoneNumberA), check(NameCP), check(UsernameCP), check(TradeWebsite), check(TransActionDate), check(PurchaseMoney), check(PaymentMethod), check(BankAccountCP), check(AccountHolderCP), check(TransactionTime), check(BankAccountA), check(ProductType) \rightarrow Complete)$

62. P_{int}: $finish(Complete)$

63. P_{int}: $argue(pCrime, Complete \rightarrow Intake)$

64. P_{int}: $finish(Intake)$

C Case file dialogue

C.1 Main dialogue

1. P_{ana}: $begin(Casefile)$

2. P_{ana}: $request(\text{'Toys4ever'} \wedge \text{'marktplaats.nl'})$

3. $P_{inq.m}$: $affirm(\text{'Toys4ever'} \wedge \text{'marktplaats.nl'})$

4. $P_{inq.m}$: $challenge(relevant(\text{'Toys4ever'} \wedge \text{'marktplaats.nl'}))$

5. P_{ana}: $argue(A_{relevant1} = (((((((OfferedByCP, (\neg DeliveredByCP, Waited5Days \Rightarrow \neg SentByCP), (TrackAndTrace \Rightarrow LegitimateSalesIndicator)) \Rightarrow DeceitfulSalesmanCP) \Rightarrow FraudScheme), (OfferedByCP, OrderedByA \Rightarrow MutualAgreement), ((ContactThroughTradewebsite \Rightarrow ContactPrior), (\neg RespondMessageCP \Rightarrow \neg ContactAfter) \Rightarrow BreakingContact) \Rightarrow IntendedFraudulentBenefit), (OrderedByA, PaymentByA \Rightarrow ObtainingGoods)) \Rightarrow pFraud), (observed(A_{pFraud}, R_1)), (r_f1 : pFraud \rightarrow pCrime)), (observed(instance(Tradewebsite, \text{'marktplaats.nl'}) \wedge instance(UsernameCP, \text{'Toys4ever'}), R_1)) \Rightarrow relevant(\text{'Toys4ever'} \wedge \text{'marktplaats.nl'}))$

6. $P_{inq.m}$: $concede(A_{relevant1})$

...

16. $P_{inq.m}$: $report(A_{information1})$

17. P_{ana}: $concede(A_{information1})$

18. $P_{inq.b}$: $deny(\text{'Toys4ever'} \wedge \text{'marktplaats.nl'})$

19. P_{ana}: $request(\text{'NL00CASH9999'})$

20. $P_{inq.m}$: $deny(\text{'NL00CASH9999'})$

21. $P_{inq.b}$: $affirm(\text{'NL00CASH9999'})$

22. $P_{inq.b}$: $challenge(relevant(\text{'NL00CASH9999'}))$

23. P_{ana}: $argue(A_{relevant2} = (((((((OfferedByCP, (\neg DeliveredByCP, Waited5Days \Rightarrow \neg SentByCP), (TrackAndTrace \Rightarrow LegitimateSalesIndicator)) \Rightarrow DeceitfulSalesmanCP) \Rightarrow FraudScheme), (OfferedByCP, OrderedByA \Rightarrow MutualAgreement), ((ContactThroughTradewebsite \Rightarrow ContactPrior), (\neg RespondMessageCP \Rightarrow \neg ContactAfter) \Rightarrow BreakingContact) \Rightarrow IntendedFraudulentBenefit), (OrderedByA, PaymentByA \Rightarrow ObtainingGoods)) \Rightarrow pFraud), (observed(A_{pFraud}, R_1)), (r_f1 : pFraud \rightarrow pCrime)), (observed(instance(BankAccountCP, \text{'NL00CASH9999'}), R_1)) \Rightarrow relevant(\text{'NL00CASH9999'}))$

24. $P_{inq.b}$: $concede(A_{relevant2})$

...

34. $P_{inq.b}$: $report(A_{information2})$

35. P_{ana}: $concede(A_{information2})$

36. P_{ana}: $argue(A_{Casefile})$

36. P_{ana}: $finish(Casefile)$

C.2 Dialogue with Marktplaats agent)

7. $P_{inq.m}$: $begin(information(\text{`Toys4ever'} \wedge \text{`marktplaats.nl'}))$

8. $P_{inq.m}$: $inquire(\text{`Toys4ever'} \wedge \text{`marktplaats.nl'})$

9. $P_{ext.m}$: $affirm(\text{`Toys4ever'} \wedge \text{`marktplaats.nl'})$

10. $P_{ext.m}$: $challenge(relevant(\text{`Toys4ever'} \wedge \text{`marktplaats.nl'}))$

11. $P_{inq.m}$: $argue(A_{relevant1})$

12. $P_{ext.m}$: $concede(A_{relevant1})$

13. $P_{ext.m}$: $report(A_{information_1} = (((\neg known(NameRegistered)) \rightarrow check(NameRegistered)), ((\neg known(AddressRegistered)) \rightarrow check(AddressRegistered)), ((\neg known(AddressRegistered)) \rightarrow check(ResidenceRegistered)), (((instance(EmailAddressRegistered, \text{e-mail address 1})) \rightarrow known(EmailAddressRegistered)) \rightarrow check(EmailAddressRegistered)), ((\neg known(PhoneNumberRegistered)) \rightarrow check(PhoneNumberRegistered)), ((instance(BankAccountRegistered, \text{`NL00BANK1234'}) \rightarrow known(BankAccountRegistered)) \rightarrow check(BankAccountRegistered)), ((instance(IPAddressRegistered, \text{IP address 1}) \rightarrow known(IPAddressRegistered)) \rightarrow check(IPAddressRegistered)) \Rightarrow information(\text{`Toys4ever'} \wedge \text{`marktplaats.nl'})))$

14. $P_{inq.m}$: $concede(A_{information_1})$

15. $P_{inq.m}$: $finish(information(\text{`Toys4ever'} \wedge \text{`marktplaats.nl'}))$

C.3 Dialogue with Bank agent

25. $P_{inq.b}$: $begin(information(\text{`NL00BANK1234'}))$

26. $P_{inq.b}$: $inquire(\text{`NL00BANK1234'})$

27. $P_{ext.b}$: $affirm(\text{`NL00BANK1234'})$

28. $P_{ext.b}$: $challenge(relevant(\text{`NL00BANK1234'}))$

29. $P_{inq.b}$: $argue(A_{relevant2})$

30. $P_{ext.b}$: $concede(A_{relevant2})$

31. $P_{ext.b}$: $report(A_{information_2} = (((instance(NameRegistered, \text{`Dennis de Boer'}) \rightarrow known(NameRegistered)) \rightarrow check(NameRegistered)), ((instance(AddressRegistered, \text{address 1}) \rightarrow known(AddressRegistered)) \rightarrow check(AddressRegistered)), (((instance(ResidenceRegistered, \text{residence 1}) \rightarrow known(ResidenceRegistered)) \rightarrow check(ResidenceRegistered)), ((instance(EmailAddressRegistered, \text{email address 2}) \rightarrow known(EmailAddressRegistered)) \rightarrow check(EmailAddressRegistered)), ((instance(PhoneNumberRegistered, \text{phone number 1}) \rightarrow known(PhoneNumberRegistered)) \rightarrow check(PhoneNumberRegistered)), ((instance(BankAccountRegistered, \text{`NL00BANK1234'}) \rightarrow known(BankAccountRegistered)) \rightarrow check(BankAccountRegistered)), ((\neg known(IPAddressRegistered)) \rightarrow check(IPAddressRegistered)) \Rightarrow information(\text{`NL00BANK1234'})))$

32. $P_{inq.b}$: $concede(A_{information_2})$

33. $P_{inq.b}$: $finish(information(\text{`NL00BANK1234'}))$

C.4 Concluding argument $A_{Casefile}$

A_1 : $instance(NameRegistered, \text{`Dennis de Boer'})$

A_2 : $instance(NameCP, \text{`Dennis de Boer'})$

A_3 : $instance(NameCP, \text{`Lukas Goedeman'})$

A_4 : $A_1 \Rightarrow instance(NameSuspect, \text{`Dennis de Boer'})$

A_5 : $A_1, A_3 \Rightarrow instance(FakeName, \text{`Lukas Goedeman'})$

A_6 : $instance(AddressRegistered, \text{address 1})$

A_7 : $\neg known(AddressCP)$

A_8 : $A_1 \Rightarrow instance(AddressSuspect, \text{address 1})$

A_9 : $A_2 \Rightarrow \neg known(FakeAddress)$

A_{10} : $instance(ResidenceRegistered, \text{residence 1})$

A_{11} : $\neg known(ResidenceCP)$

A_{12} : $A_{10} \Rightarrow instance(ResidenceSuspect, \text{residence 1})$

A_{13} : $A_{11} \Rightarrow \neg known(FakeResidence)$

A_{14} : $instance(EmailAddressRegistered, \text{email address 1})$

A_{15} : $instance(EmailAddressRegistered, \text{email address 2})$

A_{16} : $A_{14} \Rightarrow instance(EmailAddressSuspect, \text{email address 1})$

A_{17} : $A_{15} \Rightarrow instance(EmailAddressSuspect, \text{email address 2})$

A_{18} : $instance(PhoneNumberRegistered, \text{phone number 1})$

A_{19} : $A_{18} \Rightarrow instance(PhoneNumberSuspect, \text{phone number 1})$

A_{20} : $instance(BankAccountRegistered, \text{'NL00BANK1234'})$

A_{21} : $instance(BankAccountCP, \text{'NL00BANK1234'})$

A_{22} : $A_{20} \Rightarrow instance(BankAccountSuspect, \text{'NL00BANK1234'})$

A_{23} : $A_{20}, A_{21} \Rightarrow \neg known(MoneyLaundering)$

A_{24} : $instance(IPAddressRegistered, \text{IP address 1})$

A_{25} : $A_{24} \Rightarrow instance(IPAddressSuspect, \text{IP address 1})$

A_{26} : $A_4 \Rightarrow identity(NameSuspect, \{\text{'Dennis de Boer'}\})$

A_{27} : $A_8 \Rightarrow identity(AddressSuspect, \{\text{address 1}\})$

A_{28} : $A_{12} \Rightarrow identity(ResidenceSuspect, \{\text{residence 1}\})$

A_{29} : $A_{16}, A_{17} \Rightarrow identity(EmailAddressSuspect, \{\text{email address 1}, \text{email address 2}\})$

A_{30} : $A_{19} \Rightarrow identity(PhoneNumberSuspect, \{\text{phone number 1}\})$

A_{31} : $A_{22} \Rightarrow identity(BankAccountSuspect, \{\text{'NL00BANK1234'}\})$

A_{32} : $A_{25} \Rightarrow identity(IPAddressSuspect, \{\text{IP address 1}\})$

$A_{Identity}$: $A_{26}, A_{27}, A_{28}, A_{29}, A_{30}, A_{31}, A_{32} \Rightarrow Identity$

A_{33} : $A_5 \Rightarrow identity(FakeName, \{\text{'Lukas Goedeman'}\})$

A_{34} : $A_9 \Rightarrow identity(FakeAddress, \emptyset)$

A_{35} : $A_{13} \Rightarrow identity(FakeResidence, \emptyset)$

A_{36} : $A_{23} \Rightarrow identity(MoneyLaundering, \emptyset)$

$A_{FakeIdentity}$: $A_5, A_9, A_{13}, A_{23} \Rightarrow FakeIdentity$

A_{37} : $OfferedByCP$

A_{38} : $\neg DeliveredByCP$

A_{39} : $Waited5Days$

A_{40} : $A_{38}, A_{39} \Rightarrow \neg SentByCP$

A_{41} : $TrackAndTrace$

A_{42} : $ProductPictured$

A_{43} : $A_{41}, A_{42} \Rightarrow LegitimateSalesIndicator$

A_{44} : $A_{37}, A_{40}, A_{43} \Rightarrow DeceitfulSalesmanCP$

A_{45} : $A_{44} \Rightarrow FraudScheme$

A_{46} : $OrderedByA$

A_{47} : $A_{37}, A_{46} \Rightarrow MutualAgreement$

A_{48} : $ContactThroughTradewebsite$

A_{49} : $A_{48} \Rightarrow ContactPrior$

A_{50} : $\neg RespondMessageCP$

A_{51} : $A_{50} \Rightarrow \neg ContactAfter$

A_{52} : $A_{49}, A_{51} \Rightarrow BreakingContact$

$$A_{53}: A_{45}, A_{47}, A_{52} \Rightarrow IntendedFraudulentBenefit$$
$$A_{54}: PaymentByA$$
$$A_{55}: A_{46}, A_{54} \Rightarrow ObtainingGoods$$
$$A_{56}: A_{53}, A_{55} \Rightarrow pFraud$$
$$A_{pCrime}: A_{56} \rightarrow pCrime$$
$$A_{Casefile}: A_{Identity}, A_{FakeIdentity}, A_{pCrime} \rightarrow Casefile$$