Department of Computer Science
Technical University of Cluj-Napoca

Logic-based Models of Argument

Habilitation thesis

Adrian Groza

Contents

Part I

Preliminaries

Chapter 1

Advising and management activities

There have been 11 years since I've been accepted as a teacher in computer science, based on my inaugural thesis (inaugural dissertation). Indeed, in 2008 I defended my PhD thesis "Structural Models for Inter-agents Online Dispute Resolution". The thesis got the distinction *Magna Cum Laude*. My subjective explanations on this distinction are two: First, instead of taking six years as scheduled, the work took only five. I recall that this interval was also quite short comparing to other theses in the same period. Second, the thesis was warranted by some papers published at conferences with low acceptance rate (i.e. 15% for IAT) and many papers published in LNCS. Up to 2006, LNCS was still considered an ISI journal with impact factor. Now, I am writing this habilitation thesis (or Habilitationsschrift) aiming to prove to myself that I am capable of - how they say - "proficient transfer of knowledge to the next generation" [43].

I am an Associate Professor at Technical University of Cluj-Napoca since 2014, with experience in artificial intelligence. During my PhD (2003-2008) I taught only laboratory works. After PhD, I had lectures on Functional Programming, Artificial Intelligence, Intelligent Systems, and Knowledge-based Systems.

Interestingly, these classes have been perceived so differently by students during the years. For instance, in 2008 lambda functions or higher order functions in Haskell and Lisp were concepts classified by students as "abstract, stupid, and of no practical relevance". Now, not few students are fascinated by the same concepts in Java or Python. In 2008, multi-layer perceptrons (known today as deep neural networks) were considered by students no more than an ugly topic for the exam. Now, students are using such instrumentation - driven by their own interests - long before I teach them in the second semester of the 3^{rd} year. Differently, in 2008 ontologies were considered an hot topic. I recall a session of diploma projects with a full day booked by the ontology-related topics. Now, the tribe of knowledge representation and reasoning is (arguable) under the shadow of the statistically, black boxes tribe leaded by the shining star of deep neural networks.

I did look with a bird-eye at the developments in machine learning and even I used such instrumentation to solve tasks related to natural language processing on texts on climate change or to classify images with crops in precision agriculture. Still, I am a member of the first tribe - the knowledge-based and reasoning one. In this line, my work is on knowledge representation and reasoning, multi-agent systems, and argumentation theory.

I have acquired funding as principal investigator for the following research topics:

- ARGCLIME: Increasing understanding on climate change through public discourse analyse and stakeholders modelling, EEA and Norwegian Financial Mechanisms (3 months), 2016
- ARGSAFE: Using Argumentation for justifying safeness in complex technical systems, UEFSCDI, PN-II-Capacitati Bilateral Romania-Argentina (24 months), 2013-2015
- GREEN-VANETS: Improving transportation using Car-2-X communication and multi agent systems, Intern TUCN (12 months), 2013-2014
- LELA: Collaborative Recommendation System in the Tourism Domain Using Semantic Web Technologies and Text Analysis in Romanian Language, UEFSCDI, PN-II Innovation Checks, (6 months), 2013-2014
- ASDEC - Structural Argumentation for Decision Support with Normative Constraints, UEFSCDI, PN-II-Capacitati Bilateral Romania-Moldova (20 months), 2013-2014
- ARGNET - Structured Argumentation in Semantic Web Oriented E-business, POSDRU-EXCEL (Postdoc Research Grant), 2010-2013
- Automating Online Dispute Resolution for B2B Using Multi-Agent Systems, CNCSIS, TD-534/2007, 2007-2008

The LELA project has obtained the 1^{st} rank after the evaluation in the national competition.

I served as a reviewer to several journals such as Expert Systems with Applications, Semantic Web and Information Systems, Argument&Computation, Information Technology Research.

Recently, I was the general co-chair of The Sixth International Conference on. Mining Intelligence and Knowledge Exploration (MIKE 2018), LNAI, Springer,

2018. MIKE 2018 received 93 submissions from 29 countries. The program committee from 27 countries recommended 33 of these submissions for acceptance. Hence, the overall acceptance rate for this edition of MIKE is 35.48%. The authors represent institutions from Austria, Colombia, Ecuador, Germany, India, Italy, Japan, Mexico,Portugal, Romania, South Africa, Korea, Spain, Switzerland, Turkey, and United Arab Emirates.

I also organised the SIMA workshop (Chisinau, Moldova), collocated with ICMCS conference in 2014, and the 1st Workshop on Flexible Communication Between Human and Software Agents Cluj-Napoca, Romania, collocated with The 10th National Conference on Human-Computer Interaction (ROCHI) in 2013.

I organised three editions of the Ontology Building Competition (BOC13), (BOC14), (BOC15). The ontologies developed at these competitions were published on the OntoHub online ontology repository.

I was the project leader or developer for several scientific software, available for download:

- ARGSENSE aggregates public opinions from debate sites, 2018
- ARGMED identifies medical arguments in scientific papers, 2016
- EMKA identifies conflicting information in medical papers, 2016
- SafeEd checks consistency of GSN safety cases, 2014
- OntEval evaluates ontologies with Analytical Hierachical Process, 2014
- OntoRich - automatic ontology enrichment, 2011
- ARGNET supports argumentation in semantic wikis, 2010

Some of my research has been awarded distinctions: best paper at FEDCSIS, Warsaw, (among 150 accepted papers), 2014; UEFSCDI award for the paper - Compliance checking of integrated business processes. Data & Knowledge Eng., 87, 1-18, 2014; UEFSCDI award for Assuring safety in air traffic control systems with argumentation and model checking. ESWA, 44, 367-385, 2016. I also obtained several travel grants: travel grant at Poznan Reasoning Week, Adam Mickiewicz University and University of Zielona Gora., Poznan, Poland, 2018, travel grant COST Act. Agreement Tech. King's College, London, 2012; travel grant European Assoc. of Artif. Intell., TUHH, Hamburg, 2008.

1.1 Teaching activities

My teaching activities started in 2003 at the Technical University of Cluj-Napoca with the practical classes for *Multi-Agent Systems* and *Artificial Intelligence*. After

Ph.D. defending in 2008, I started lectures on *Introduction to Artificial Intelligence*, *Intelligent Systems* and *Functional Programming*. I was responsible for creating the content and teaching the newly introduced *Knowledge-Based Systems*.

I had recently (May 2018) 6 lectures at the Department of Artificial Intelligence and Technical University of Kosice, Slovakia. I has also lectures the summer school *Electrosummer*, 15-28 Iulie 2018, Cluj-Napoca. The lectures were on Description Logics and Epistemic Logic.

I have supervised 16 master students (since 2011), respectively 57 undergraduate students (since 2010). I lead the students towards scientific research and writing. In this line, 14 such students have participated at different editions of the Computer Science Student Conference . The conference is co-organised each year by the Department of Computer Science at Babes-Bolyai University and the Department of Computer Science at TUCN. The students obtained the following awards: 1^{st} prize (2011), 2^{nd} prize (2011), 3^{rd} prize (2012), 1^{st} prize (2014), 2^{nd} and 3^{rd} prize (2015), 3^{rd} prize (2016), 1^{st} prize (2017).

I introduced the *AI news* activity in the first 6 minutes of the lecture. Here, three students present some news on AI. The focus is on technologies behind the recent developments of artificial intelligence and how these technologies are projected in the public arena. Beside the warm-up goal, the activity also stimulates critical thinking at the students. This critical thinking is relevant in context of deceptive information largely available online on computer related issues. The students realise the flaws in the discourse or the exaggerated business style when presenting minor conceptual achievements.

Following the experience at Norwegian University of Science and Technology, Trondheim, Norway, (where most of the lectures do end with a Kahoot game), I introduced the Kahoot game at Technical University of Cluj-Napoca in 2016. The game takes 5 minutes at the end of each class. It provides statistical information on what part of the lecture was correctly/wrongly or easily/difficultly acquired by the students. The game allows me to cover in the last 5 minutes the issues that were not correctly perceived by the majority of the students. Now, the game is used by my colleges in several classes like: Introduction to Artificial Intelligence, Intelligent Systems, Functional Programming, Knowledge-based systems.

1.2 Research activities

Technical challenges that I was interested after my PhD (2009-2018) are:

- How can arguments conveyed in public arena can be aggregated? [70]
- Which are the strategic decisions in games with incomplete information? [148]

- How can improve version spaces algorithm using expert knowledge? [74]
- How can one use ontologies to develop more flexible chatbots? [161]
- How can ontologies improve the performance of textual entailment? [160]
- How can solve conflicts in ensemble learning using argumentation? [31]
- How can repair a model via model checking and argumentation? [49, 53]
- How can increase consensus between agents with different knowledge? [69]
- How can extract arguments from scientific articles? [71]
- How can agent technologies can be used in context of vehicular networks? [24]
- How to assess students essays using textual entailment and ontologies? [73]
- How to model an ontology for wind energy systems? [62]
- How can extract information from folktales [64]
- Can argumentation frameworks be solved using bipartite graphs? [75]
- Can reasoning in description logic be used for formal verification? [56]
- How can analytical hierarchy process can be used for ranking ontologies? [55]
- How to query the models in Goal Structuring Notation? [68]
- How to assess the quality of an ontology? [89]
- How ontologies can guide NLP to extract information from folktales? [159]
- How to reason on data from vehicular networks? [59]
- How can translate from Romanian to SPARQL? [125]
- How safeness of overtaking can be assure by multi-agent cooperation? [65]
- How safeness can be increasing by means of argumentation? [50]
- How can motivate students to learn ontologies by ontology competition? [60]
- How can analyse and query data streams? [10, 83]
- How can represent contracts in GoogleDocs and monitor them? [169]

- How to check business process compliance against quality standards? [115]
- How to reason on stream with plausible logic? [67, 112]
- How to justify commitment breaches? [114]
- How to automatically enrich ontologies? [63, 11]
- How to distinguish between argument and explanations in dialog? [113]
- How to distinguish between argument and justification in dialog? [111]
- How to repair ontologies using argumentation? [72]
- How to support technical audit through argumentation? [117]
- How to integrate knowledge and sentiments for a recommender system? [167]
- How to model arguments in fuzzy description logic? [58, 105, 61]
- How to enrich ontologies using semantic wikis and design patterns? [48]
- How to check conformance in the construction domain? [58]
- How to structure arguments in social media? [87, 86]
- How to check norm compliance in supply chains? [57]
- How to represent safety standards with ontologies? [108, 106, 107]
- How to optimise public transportation through learning? [118]
- How to introduce context in the Argument Interchange Format? [104]

Acknowledgement Chapter 2 is based on [31]. Chapter 3 is an extension of [103]. The work in Chapter 4 aggregates ideas published in [110, 109, 105, 61]. Chapter 5 is an extension of [70] that is under evaluation at COMSIS journal. The ideas in chapter 6 have been published in [100]. The work in chapter 7 has been published in [56, 49, 53]. Chapter 8 summarises ideas published in [116].

I acknowledge my colleges and students for their contribution to the work presented here: Radu Razvan Slavescu, Anca Marginean, Stefan Contiu, Anca Goron, Pinar Ozturk, Sergiu Gomez, Ioan Alfred Letia.

Table 1.1: Conceptual instrumentation investigated (2009-2018).

Description Logics	[115], [161], [160], [73], [62], [59], [60], [169] [64], [56], [55], [68], [63, 11], [113], [72], [117], [167], [58, 105, 61], [48], [58], [87, 86], [104]
Natural language processing	[161], [160], [69], [71], [73], [64], [125]
Machine learning	[31], [74], [118]
Fuzzy logic	[74], [69], [58, 105, 61]
Justification logic	[114], [113]
Defeasible logic	[31], [49, 53], [50]
Stream Reasoning	[10, 83], [67, 112]
Probabilities	[148]
Model checking	[49, 53],[56], [68], [115]
Multi-agent systems	[69], [24], [65], [113]

Table 1.2: Application domains investigated (2009-2018).

Agriculture	[31], [74]
Medical	[69], [71], [50], [10, 83]
Food	[115], [67, 112], [58, 105, 61]
Autonomous vehicles	[49, 53], [24],[56], [68], [59], [65], [118]
Climate change	[70], [161], [160], [62]
Tourism	[125], [63, 11], [167]
Law	[115], [169], [114], [113], [113], [117], [58]
Education	[73], [55], [60]

Part II

Logic-based argumentation

Chapter 2

Arguing in defeasible logic

> "How can you not transform if you can transform"
>
> ---
>
> Stanislaw Lem

2.1 Conflict resolution in ensemble learners

The ideas in this chapter were presented in [31]. Ensemble learners have already been proved to be more effective than single learning models, especially when the correlation of the errors made by the base learners is low [97]. Such low correlation leads to conflictual decisions among base classifiers. Conflicts within an ensemble learners is usually solved with vote-based methods. These methods have difficulties in classifying border line instances correctly and also to justify their decision. The conceptual research question is on *conflict resolution in ensemble learning.* To handle debatable instances in ensemble learning and to increase transparency in such debatable classification decisions, our hypothesis is that argumentation could be more effective than voting-based methods [31].

The contribution in [31] was that voting system in ensemble learning is replaced by an argumentation-base conflict resoluter. Prospective decisions of base classifiers are presented to an argumentative system. This argumentative system uses defeasible logic to reasoning on pros and cons against a classification decision. The system computes a recommendation considering both the rules extracted from base learners and the available expert knowledge.

Our work here is on *conflict resolution in ensemble learning.* We have used argumentation systems on top of ensemble learning, to deal with disputed classes. We have extract the classification knowledge encapsulated by each learner. The

Table 2.1: Features of the crop dataset obtained from Landsat 8.

Landsat 8 OLI Band	Feature Name	Justification
3 (Green)	Green Level	Indicates peak vegetation.
6 (Short-wave infrared)	Moisture Level	Indicates moisture content of both soil and vegetation.
4 (Red) and 5 (Near infrared)	Normalized Difference Vegetation Index	Indicates photosynthetic activity.

aim was to apply argumentation framework to classification, following the research direction opened by [4, 174, 80].

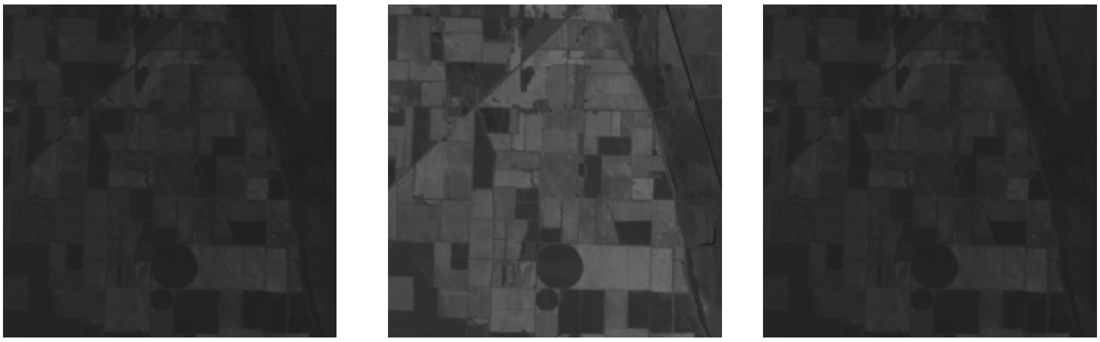
2.2 Running scenario: crop classification

We illustrate the method on a decision support system for crop classification into four classes: corn, soybean, cotton, and rice. The input data is taken from satellite images. We wanted to develop a method that can benefit from both huge raw data extracted from satellite images, but also from the robust amount of expert knowledge for agriculture. Thus, we developed a hybrid intelligent system that can exploit both agricultural expert knowledge and machine learning. As the crop raw data is characterized by heterogeneity, we have used ensemble learners, while expert knowledge is encapsulated within a rule-based system.

The test site used for our classification experiments is an area of 20 square kilometers in the New Madrid County, southeast of the Missouri State, USA. This site has a humid subtropical climate and favorable agricultural activities, with an average of 1,087 acres per farm land of which 96.5% is used as cropland [164].

The Landsat image was acquired on July 5th, 2014. and exported into GeoTIFF format by using the USGS online system[1]. A Landsat image consists of multiple grayscale 16-bit images, each storing a spectral band captured by the satellite. Four out of the nine OLI (Operational Land Imager) bands are used for constructing the classification data-set. The four bands are chosen based on their correlation to the vegetation discrimination process. Table 2.1 lists the four bands together with the extracted features. Bands 3 and 6 are used as features in their raw format, while bands 4 and 5 are combined into a new feature: Normalized Difference Vegetation Index (NDVI). NDVI is a proven indicator of land-use and cover changes [36], being calculated from the red (*Red*) and near infrared bands (*NIR*) by the following formula: $NDVI = (NIR - Red)/(NIR + Red)$. Figure 2.1

[1] http://earthexplorer.usgs.gov/



(a) Green (Band 3) (b) Moisture (Band 6) (c) NDVI (Bands 4 and 5)

Figure 2.1: Landsat gray-scale images corresponding to the features set.

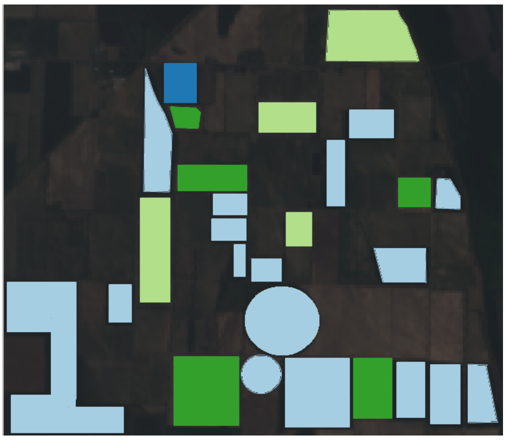


Figure 2.2: Landsat RGB image with ground truth mask (highlighted pixels per crop class). The color codes for *corn*, *rice*, *cotton* and *soybean* are light blue, blue, light green and green respectively. The dataset contains 5,407 instances.

displays the Landsat gray-scale images corresponding to the three features of the dataset: green, moisture and NDVI.

The ground truth reference is obtained from the US Department of Agriculture - Statistics Service[2], using a filter for the area of interest and the timestamp of the Landsat image. The resulting ground truth image is depicted in figure 2.2.

The obtained classification dataset is split into two independent datasets: 20% used for training and validating the classification models and 80% used for testing. This split mimics the idea that the classifiers should be able to predict vast areas after being trained on a small number of plots, a characteristic of the crop classification problem, as obtaining ground truth references is often associated with the

[2]http://nassgeodata.gmu.edu/CropScape/

effort of inspecting the plots in person. The resulted training set contains 1,065 instances, while the test set contains 4,342 instances.

2.2.1 Arguing on inconsistent classification

In the case of inconsistent classifications by two or more learning algorithms, more analyze is required either by human intervention or by more accurate technical instrumentation. Ar argumentation machinery can support the decision of the human expert by providing pro and counter arguments for a debatable class. The resulted argumentation framework, complemented with human knowledge leads to justified decisions in case of class controversy among base classifiers in an ensemble.

The criteria used for deciding for which instances to accept the classification and for which to apply the argumentation machinery is when at least one base classifier outputs a different classification for the given instance.

Definition 1 *An instance i belongs to the conflict set Γ iff there are at least two learners in the ensemble $\mathcal{H}$ that output different classes for that instance i:*

$$i \in \Gamma \textit{ iff } \exists\, h_l, h_{l'} \in \mathcal{H}, l \neq l',\ \textit{s.t. } h_l(x_i) = y_i, h_{l'}(x_i) = y_i' \textit{ with } y_i \neq y_i'$$

Example 1 *Consider the binary ensemble $\mathcal{H} = \{h_{dt}, h_{nn}\}$ formed by a decision tree and a neural network classifier employed for a binary classification, $y \in \{-,+\}$. The conflict set Γ is formed by all instances that the decision tree classifies "+" and the neural network "−", together with the ones that the decision tree classifies "−" and the neural network "+". Given the labeled datasets $\mathcal{D}(h_{dt}) = \{(i_1,+),(i_2,+),(i_3,-),(i_4,-)\}$ and $\mathcal{D}(h_{nn}) = \{(i_1,+),(i_2,-),(i_3,+),(i_4,-)\}$, the conflict set would be $\Gamma = \{i_2, i_3\}$. We assume that no further analysis is required for the instances outside the conflict set Γ. Here, all learners in $\mathcal{H}$ agree on the class of the instances i_1 and i_4.*

Definition 2 *A classification rule is an implication $condition(x) \rightarrow y$ where the condition is a conjunction of tests over the features of input x and y is the class.*

Example 2 *Consider a bi-dimensional input dataset of binary values 0 and 1 that needs to be classified following the logical AND operation. The classification rule for class "+" is: $(equal(x_0, 1) \wedge equal(x_1, 1) \rightarrow " + ")$, while class "−" is described by two classification rules $(equal(x_0, 0) \rightarrow " - ")$ and $(equal(x_1, 0) \rightarrow " - ")$.*

The scope of the classification rules is to describe why a classifier $h \in \mathcal{H}$ believes that class y should be assigned to an instance x_i.

Definition 3 *An ensemble knowledge base Ens_{KB} is the set of all classification rules that describe the classification for each classifier $h \in \mathcal{H}$.*

An argumentation knowledge base merges two sources of knowledge: rules mined from base classifiers (Ens_{KB}) and expert knowledge (E_{KB}). As Ens_{KB} contains inconsistent rules, all the argumentation base will contain conflicting rules. The expert knowledge E_{KB} is defined by distinguishing different classes y based on the similar features of the classification dataset $\mathcal{D}$. The expert knowledge features can be augmented by deriving new features from existing ones. The two sources of knowledge are aggregated as a DeLP program that performs dialectical analysis to decide the class of the given instance. Formally:

Definition 4 *An argumentation knowledge base is a tuple $\mathcal{A} = \langle Ens_{KB}, E_{KB}, \oplus \rangle$, where Ens_{KB} represents the knowledge extracted from the ensemble learner and E_{KB} is the domain expert knowledge. The aggregation strategy $\oplus$ for the set $\{Ens_{KB}, E_{KB}\}$ applies the set of conflict resolution strategies (heuristics) $\mathcal{RS}$ for computing a partial order relation between rules in $\{Ens_{KB}, E_{KB}\}$.*

In the basic conflict resolution strategy of defeasible logic, strict rules are stronger than defeasible rules. We note this strategy with s_0. Two other possible conflict resolution strategies are: i) s_1: expert knowledge stronger than any classifier knowledge: $\forall\, r \in E_{KB}$ and $\forall\, s \in Ens_{KB}$, $r \succ s$ or ii) s_2: specific rules stronger than general rules: given $r : a_i \rightarrow y_1$ and $s : b_i \rightarrow y_2$ if $\{b_i\} \subset \{a_i\}$ then $r \succ s$. Hence, a possible aggregation strategy is $\oplus = [s_0, s_1, s_2]$.

Our top level approach is captured by algorithm 1. Given the ensemble of classifiers $\mathcal{H} = \{h_1, ..., h_L\}$ and an instance case x by its vector of features, the algorithm 1 outputs the class y of instance x. If all the classifiers h_i agree on the class of an individual, then that classification is returned (lines 1-2). In the case of conflict between classifiers in $\mathcal{H}$, the set of ensemble knowledge base Ens_{KB} is developed by unifying the extracted classification rules from all base classifiers (lines 4-7). The method ExtractRules has specific implementation for each base classifier. DeLP reasoner is asked to produce a *Undefeated* (*True*) or *Defeated* (*False*) answer for each class $y \in \{1..K\}$, by using the ensemble classification Ens_{KB} and expert rules E_{KB} as knowledge bases (lines 8-10). If there exists exactly one class that receives a *True* answer from the DeLP reasoner, then this class settles the dispute (lines 11-12). Otherwise, the classification is undecided (line 14).

2.2.2 System architecture

The system architecture is presented in Figure 2.3. The top level encapsulates data layer operations. The area of interest is extracted from the input satellite image. The features of the classification dataset are extracted from the multispectral values. The obtained dataset is normalized and split into two sets, one used for training the base classifiers and one for validating the ensemble learner.

Algorithm 1: Classifying a new instance case.

Input: $\mathcal{H} = \{h_1, ..., h_L\}$, ensemble of classifiers h_l, $l \in \{1..L\}$
Input: x, feature vector of the new case
Input: E_{KB}, expert knowledge
Output: y, class of the new case, $y \in \{1..K\}$
Output: $\mathcal{T}$, dialectical tree

```
if ∀ h_l ∈ H, h_l(x) = y then
    return y
else
    Ens_KB ← {}
    foreach classifier h_l ∈ H do
        rules ← ExtractRules(h_l(x))
        Ens_KB ← Ens_KB ∪ {rules}
    end
    answer ← {}
    foreach class y ∈ {1..K} do
        answer_y ← DelpAnswer(KnowledgeBase: Ens_KB ∪ E_KB, Query: y?)
    end
    if ∃!y ∈ {1..K} s.t. answer_y=true and ∀ z ≠ y answer_z ≠ true then
        return y, T
    else
        return undecided, T
    end
end
```

The middle layer covers the three independent classifiers: decision tree, artificial neural network and support vector machine, that compose the statistical ensemble learner $\mathcal{H} = \{h_{dt}, h_{nn}, h_{svm}\}$. The base classifiers are trained and tested by using inputs only from the training set. Each trained classifier is asked to predict the class of instances in the validation set, together with argumentation rules.

The bottom layer encloses the argumentation framework used in case of conflicts among the learners from the ensemble $\mathcal{H}$. The inputs of this layer are the classification rules extracted from each classifier. The rules are merged with expert defined knowledge and are sent to a DeLP reasoner for conflict resolution.

2.3 Interleaving rule mining and agricultural knowledge

This section covers the knowledge part used for the crop classification. Firstly, the section presents the methods for extracting rules from the three statistical classifiers: decision tree, neural network and support vector machine. Secondly, the strategy for building the expert knowledge is detailed, together with concrete samples of derived expert rules.

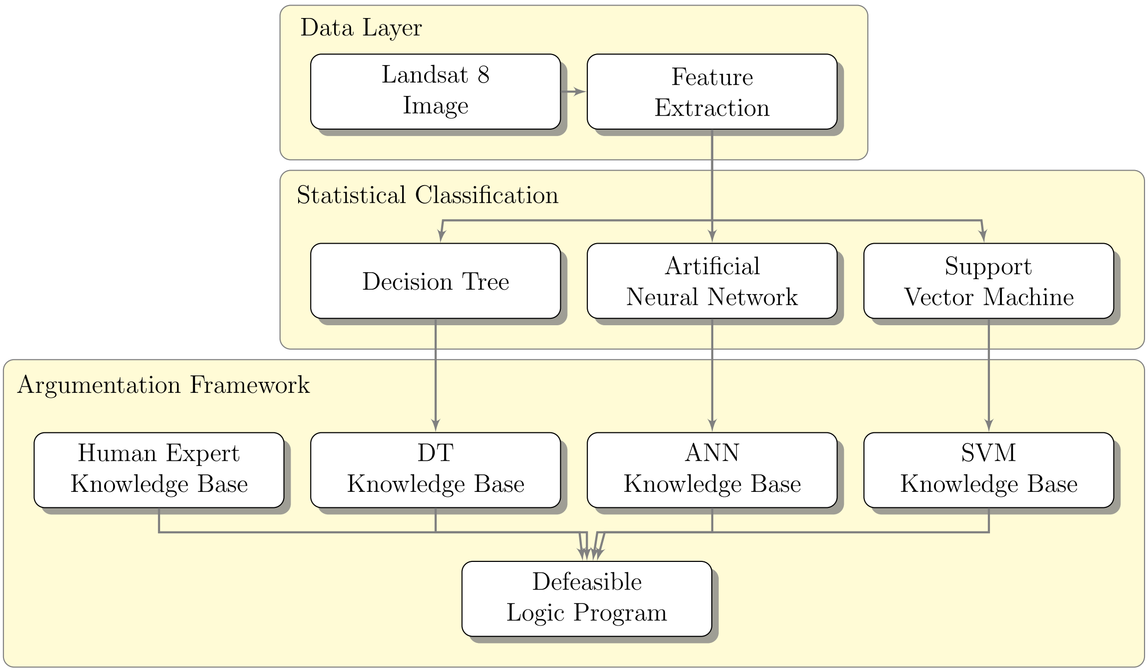


Figure 2.3: Classification system architecture.

2.3.1 Extracting DeLP rules from base learners

This section introduces the proposed methods for extracting DeLP rules from an ensemble learner. Let $\mathcal{H} = \{h_{dt}, h_{nn}, h_{svm}\}$ and classes $y = \{corn, rice, cotton, soybean\}$.

Generating defeasible rules from decision tree classifier. In a decision tree, each non-leaf node represents a condition of the form: $x_i < threshold$, where x_i is a feature of the dataset while each leaf node denotes a class. Optimal features and threshold values are determined by using the CART (Classification and Regression Trees) algorithm [19] which maximizes the information gain for each node. The 10-fold cross validation method is used for assessing the best criterion and strategy for splitting the nodes of the decision tree. All combinations of split criteria (gini or entropy) and split strategies (best or random) produced the same cross validation accuracy scores of 99.7 (+/- 0.9). Therefore, the chosen parameter values for criterion and strategies are set to gini and best split.

Translating decision tree classification rules into DeLP rules for an instance classified as y is performed by the following steps of the $\textsc{ExtractRules}(h_{dt})$ method: 1) express the branch of the tree that determined the classification as a

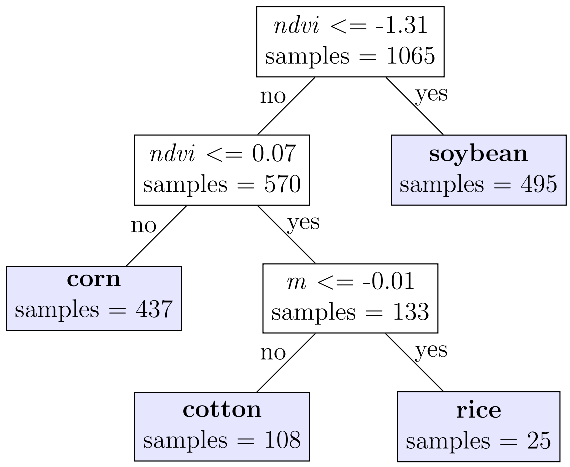


Figure 2.4: Decision tree obtained with the CART algorithm.

conjunction of conditions: $\mathcal{C} = condition_1, condition_2, ..., condition_n$; 2) introduce one defeasible DeLP rule of the form: $y \prec \mathcal{C}$; 3) introduce DeLP rules for all the other classes $y' \neq y$ of the form: $\sim y' \prec \mathcal{C}$. The rationale of the translated rules is that, given the conjunction of conditions $\mathcal{C}$, y is chosen as predicted class as long as nothing is posed against it. Similarly, any $y' \neq y$ is an incorrect prediction as long as $\mathcal{C}$ is not defeated.

Example 3 *Let the classification of cotton instances in Fig. 2.4. The cotton tree branch is expressed as a conjunction of conditions (step 1):*

$$decision\_tree(g, m, ndvi) \quad \prec \quad m > -0.01, ndvi \in [-1.31, 0.07].$$

The DeLP rule pleading for cotton class is then introduced (step 2):

$$cotton \quad \prec \quad decision\_tree(g, m, ndvi).$$

The defeasible rules which signal that all the other classes are incorrect predictions are introduced (step 3):

$$\begin{array}{rcl} \sim corn & \prec & decision\_tree(g, m, ndvi). \\ \sim rice & \prec & decision\_tree(g, m, ndvi). \\ \sim soybean & \prec & decision\_tree(g, m, ndvi). \end{array}$$

Generating defeasible rules from neural network classifier. We adopt a feed-forward neural network model containing a single hidden layer. The input layer contains three units corresponding to the input features green level, moisture level, and NDVI. The output layer contains four units corresponding to the classes of crops we intend to discriminate: corn, rice, cotton, and soybean. The predicted

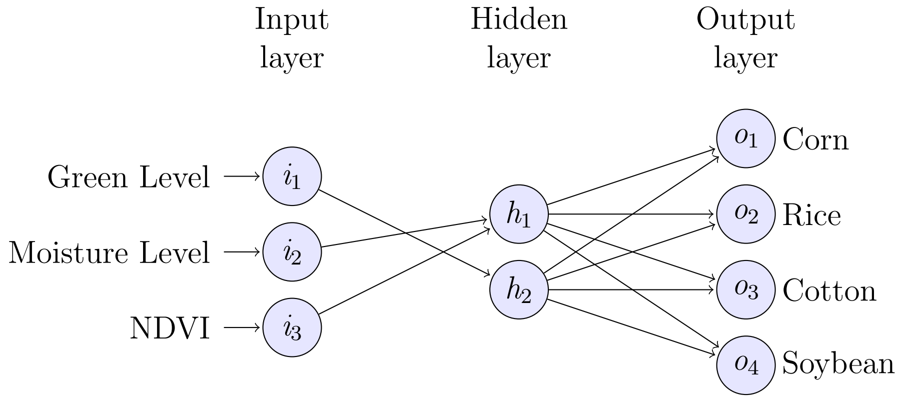


Figure 2.5: The topology of the pruned artificial neural network.

class is determined by the output layer unit having the maximum value. The 10-fold cross validation method is used to determine the number of units in the hidden layer. Experiments with more than two hidden units show no growth from 98.4(+/-2.4) averaged accuracy across the ten folds, therefore the network is set with two nodes in the middle layer.

Skeletal pruned neural networks produce simpler classification rules [154] thus, the redundant connections between the input and hidden layers are removed. Input layer contains three nodes, while hidden layer contains two nodes. Hence, there are six possible directed edges from the input to the hidden layer. Considering that any edge of the six can be on or off, there are $2^6 = 64$ different ways of connecting the input and the hidden layers. Each of the 64 network configurations is trained and evaluated using 10-fold cross validation. The selected optimal configuration has the least number of edges from input to hidden layers and a maximal accuracy. The network with three edges between input and hidden layers depicted in figure 2.5 has achieved a maximal accuracy of 98.4(+/-2.4) on the crops training dataset.

The hyperbolic tangent function is used instead of the standard logistic function as an activation method due of faster convergence [101]. The hyperbolic tangent function is symmetric to the origin, similarly with the training dataset, on which we perform Gaussian normalization during the features extraction phase.

The neural network is trained by using the backpropagation algorithm in conjunction with gradient descent. Learning rate and momentum are set to 0.5 and 0.01 respectively by using a trial-and-error method. The network is trained for a maximum number of 2000 epochs.

Once the network is trained to optimal accuracy, the set of classification rules is extracted. The neural network will use these rules as arguments when asked by the DeLP reasoner to explain its classification. The extraction steps are based on the NeuroLinear algorithm, used for extracting oblique decision rules from trained neural networks [154]. This method is complemented with a CART decision tree classifier [19] and with a translation of decision rules to defeasible logic rules. The steps of the ExtractRules(h_{nn})) method are as follows:

Step 1. Start by training the pruned neural network depicted in figure 2.5, which consists of i_1, i_2, i_3 input units, h_1, h_2 hidden units and o_1, o_2, o_3, o_4 output units. The network has three edges between the input and hidden layers, their optimal weights being determined during training w_1 (from i_1 to h_2), w_2 (from i_2 to h_1) and w_3 (from i_2 to h_2).

Step 2. Re-pass all dataset inputs through the trained neural network and collect the values of the two hidden nodes h_1 and h_2. Hence, a new bi-dimensional dataset H is obtained.

Step 3. Train a decision tree learner by using the CART algorithm on the new bi-dimensional dataset H and obtain a set of classification rules. These rules are expressed in terms of the hidden nodes values h_1 and h_2.

Step 4. Generate rules that are expressed in terms of the input features i_1, i_2, i_3. First, hidden nodes are expressed in terms of the input nodes as $h_1 = i_2 * w_2 + i_3 * w3$ and $h_2 = i_1 * w_1$. Second, as the hidden units are the result of *tanh* function, the inverse function $tanh^{-1}$ is applied on the rules decision boundaries.

Step 5. Translate the decision rules to DeLP statements by using the same steps described for the decision tree classifier ExtractRules(h_{dt}).

Example 4 *Consider the process of extracting classification rules for the soybean class from the neural network in figure 2.5. After collecting all the neural network activation values (steps 1 and 2) and applying the decision tree classifier (step 3), the decision rule for soybean class is determined to be (step 4):*

$$h_1 > 1.83 = tanh^{-1}(0.95)$$

where h_1 is a hidden node connected to the moisture level(m) and ndvi nodes by edges with weights 0.21 and -1.44 respectively. The classification rule can be re-written as (step 5):

$$0.21 * m - 1.44 * ndvi > 1.83$$

The DeLP rules extracted from the neural network for the soybean class are:

$$\begin{array}{rcl} neural\_net(g, m, ndvi) & \prec & 0.21 * m - 1.44 * ndvi > 1.83. \\ \sim corn & \prec & neural\_net(g, m, ndvi). \\ \sim rice & \prec & neural\_net(g, m, ndvi). \\ \sim cotton & \prec & neural\_net(g, m, ndvi). \\ soybean & \prec & neural\_net(g, m, ndvi). \end{array}$$

Generating defeasible rules from support vector machine classifier. The support vector machine (SVM) is chosen as a base classifier as it can offer a different perspective on the decision boundaries between the four classes.

Since SVM conceptually works with binary classification, a strategy needs to be employed for solving the four class (corn, rice, cotton, and soybean) task classification. "One against one" strategy is chosen, as it has been proven more suitable for practical use than "one-against-all" or DAGSVM methods [84]. In the "one against one" strategy, one SVM is built for each pair of classes. That is N(N - 1)/2 SVMs are constructed for N classes. In our case, six classifiers are constructed for a four-class classification.

A 10-fold cross validation is run on the training dataset for the following kernels: RBF, polynomial, and linear. The results 99.9(+/-0.3) for linear kernel, 99.8(+/-0.3) for RBF, and 99.3(+/-1.5) for polynomial indicate that the linear kernel is a suitable choice for the SVM model.

Rule extraction is performed by a learning-based decompositional algorithm [154] complemented with a CART [19] classifier to extract if-then-else classification rules. Decision rules are then translated to DeLP rules such that the argumentation framework can make use of them. The steps for the EXTRACTRULES(h_{svm}) method are:

Step 1. Train a linear SVM model on the input dataset.

Step 2. Identify the dataset instances which are chosen by SVM as support vectors and add them to a set V. The three-dimensional set V is a subset of the input dataset and does not include any predicted class.

Step 3. Classify V by using the same SVM model. Therefore, V is augmented with predicted classes.

Step 4. Train a decision tree learner by using the CART algorithm on the V dataset to obtain the classification rules.

Step 5. Translate the decision rules to DeLP statements by using the same steps described for the decision tree classifier EXTRACTRULES(h_{dt}).

Example 5 *Consider the classification of cotton instances of the crops dataset by using an SVM model (step 1). There are* $|V| = 25$ *dataset instances chosen to form the support vectors (step 2). After reclassifying* V *by the same SVM model (step 3) and applying a decision tree classifier (step 4), the decision rule for soybean is determined to be (step 5):* $g > 0.001$ *and* $ndvi \leqslant -1.1$*. The DeLP rules extracted from the SVM classifier explaining the soybean classification are:*

Table 2.2: Expert knowledge used for deriving expert rules. The knowledge is specific for the test site defined in section 2.2. Green, moisture, and NDVI values use the same scales as the normalized crops dataset.

	Corn	Rice	Cotton	Soybean
Green Margin	-1.13 to 0.05	-0.69 to -0.5	-0.03 to 2.48	0.21 to 3.15
Moist. Margin	-0.94 to 0.07	-1.33 to -0.51	0.18 to 2.08	0.36 - 2.31
NDVI Margin	-0.59 to 1.07	-1.37 to 0.18	-1.56 to 0.06	-2.22 to -0.28
Planting	Apr 20-May 25	May 1-May 25	May 5-May 20	May 15-Jul 1
Harvesting	Sep 20-Oct 30	Sep 25-Oct 25	Oct 5-Oct 30	Oct 10-Oct 30
Harvest Signif. Color Change	Yes	No	Yes	Yes

$$
\begin{aligned}
svm(g, m, ndvi) &\prec g > 0.001, ndvi \leqslant -1.1 \\
\sim corn &\prec svm(g, m, ndvi). \\
\sim rice &\prec svm(g, m, ndvi). \\
\sim cotton &\prec svm(g, m, ndvi). \\
soybean &\prec svm(g, m, ndvi).
\end{aligned}
$$

2.3.2 Expert knowledge

The expert knowledge is built as a subset of some of the most important morphological and phenological characteristics of the four crops. The expert knowledge is not exhaustive. Its scope is to demonstrate the feasibility of the hybrid classification method and it is derived and valid only for the area of interest. Agriculture experts should be able to refine or adapt this knowledge to other crop classification contexts. Table 2.2 lists the knowledge encapsulated by the expert system.

Each of the four crops has unique morphological and phenological characteristics. Plant morphology represents the external form and structure of the plants. Plant phenology represents the occurrence of biological events in the plant life cycle. An example of a morphological feature is the plant pigmentation which accounts for the photosynthesis function, possibly telling if the crop was dry or fresh when harvested. Examples of phenological features are date observations that can be correlated to planting and harvesting dates.

The first three rows of Table 2.2 display phenological margins of NDVI, green and moisture levels for each of the four crops. Margin values are determined from phenological profiles of sample points from an enlarged area of interest surrounding the test site. If the margins are an indicator for their class, values outside the margins deny the class. One defeasible rule is introduced for indicating the class and three defeasible rules to negate the class. The following expert rules are derived from *corn* margins, each of the other three classes produce a similar set of rules but with specific margin values:

$$\begin{array}{rcl} expert\_corn(g) & \leftarrow & g \in [-1.13, 0.05]. \\ expert\_corn(m) & \leftarrow & m \in [-0.94, 0.07]. \\ expert\_corn(ndvi) & \leftarrow & ndvi \in [-0.59, 1.07]. \\ corn & \prec & expert\_corn(g), expert\_corn(m), expert\_corn(ndvi). \\ \sim corn & \prec & \sim expert\_corn(g). \\ \sim corn & \prec & \sim expert\_corn(m). \\ \sim corn & \prec & \sim expert\_corn(ndvi). \end{array}$$

Some expert rules make use of crops planting and harvesting dates. Their reference values are displayed in the fourth and fifth rows of Table 2.2 and are extracted from the USDA Agricultural Handbook [138] for the area of interest. Because the statistical classification dataset does not make use of these features, the input dataset is augmented with values used only for the exported knowledge. Margin values are determined by an empiric method, considering that Landsat images follow a period of two weeks. The past and future images are observed, plotting the NDVI to validate the crop life time-frame and extract the approximate planting and harvesting date. Planting and harvesting rules, like margin rules, can indicate or negate a class. Examples of such expert rules derived for *corn* are:

$$\begin{array}{rcl} expert\_corn(plant) & \leftarrow & plant \in [Apr{:}20, May{:}25]. \\ expert\_corn(harvest) & \leftarrow & harvest \in [Sep{:}20, Oct{:}30]. \\ corn & \prec & expert\_corn(plant), expert\_corn(harvest). \\ \sim corn & \prec & \sim expert\_corn(plant). \\ \sim corn & \prec & \sim expert\_corn(harvest). \end{array}$$

Whether there is a significant crop color change during harvesting can be empirically correlated to the dataset by observing the decreases in the green, moisture, and ndvi features. Corn, cotton, and soybean turn into a yellow or gold color at maturity while rice still preserves a component of green. The following rules are introduced by this new feature in the set of expert rules:

$$\begin{array}{lcl} corn & \prec & harvest\_color\_change \\ \sim rice & \leftarrow & harvest\_color\_change \\ cotton & \prec & harvest\_color\_change \\ soybean & \prec & harvest\_color\_change \end{array}$$

A total of 42 strict and defeasible expert rules were derived for the four crops, as follows: 28 rules by using the marginal expert values for *green*, *moisture* and *ndvi*, 20 rules by using *plant* and *harvest* dates, and 4 rules by using the color change at the harvesting time. These rules are used for conflict resolution to improve the accuracy of our ensemble classifier.

2.4 Conflict resolution through argumentation

The section presents the DeLP argumentation mechanism for conflict resolution and exemplifies the argumentation analysis on a conflicting sample of the input dataset. We also show the experiments supporting the feasibility of our solution.

2.4.1 Resolving classification conflicts

Our method for conflict resolution makes use of an argumentative framework based on defeasible logic programming. The knowledge base of the argumentation framework is the aggregate of the statistical classifiers and expert knowledge. A defeasible logic program is constructed for each of the debatable instances. The program is asked to resolve the classification dispute and argument its decision. The following steps describe the process leading to conflict resolution:

Step 1. Add the expert generated rules to the DeLP program.

Step 2. Ask the decision tree, neural network and support vector machine for DeLP rules to provide supporting arguments for their prediction. There is no need to request the complete knowledge of the base learners since during argumentation only the reasons that led them to output conflicting classification predictions are used. Aggregate the knowledge extracted from the statistical learners with the expert knowledge into the DeLP program.

Step 3. Eliminate all mathematical formulas from the DeLP program, such that the resulted program is based solely on logic programming. All such statements are evaluated and replaced with facts.

Step 4 Query the DeLP program using each of the four crops as a query. If exactly one crop has a positive answer, then the dispute is considered settled. Otherwise, the classification is undecided.

The formal representation of the above four steps appears in algorithm 2 . During the first step (line 2) expert rules are added to the DeLP program. Since strict rules are introduced only from expert knowledge, we assume that the fact that they are noncontradictory can be validated beforehand. During the second step (lines 3-4), contradictory defeasible rules are extracted from the three statistical classifiers. In the third step (lines 5-8), all mathematical formulas are pre-processed and removed from the DeLP. In the fourth step (line 9), the constructed DeLP program is asked to produce the resolved class of the contradictory instance. The implementation of DelpResolution subroutine is detailed in algorithm 3.

Example 6 *Consider the pre-processing phase of the expert rule*

Algorithm 2: Conflict resolution using DeLP for each debatable instance.

Input: $\mathcal{H} = \{h_{dt}, h_{nn}, h_{svm}\}$, ensemble of three classifiers
Input: $\mathcal{Y} = \{corn, cotton, soybean, rice\}$, class labels
Input: Γ, conflict set of debatable instances
Input: E_{KB}, expert knowledge
Output: Y, the set of classes assigned to each instance of Γ

```
foreach x_i ∈ Γ do
    P ← E_KB
    foreach classifier h ∈ H do
        P ← P ∪ ExtractRules(h(x_i))
    end
    foreach rule r ∈ P do
        if r is a mathematical formula then
            fact ← evaluate rule r for input x_i
            P ← P \ {r} ∪ {fact}
        end
    end
    y_i ← DelpResolution(P, Y)
end
```

$$expert\_corn(plant) \leftarrow plant \in [Apr{:}20, May{:}25]$$

If the disputed instance plant value is May 10, the rule will be replaced by the fact:

$$expert\_corn(plant) \leftarrow true$$

If the disputed instance plant value is June 20, the rule will be replaced by the fact:

$$\sim expert\_corn(plant) \leftarrow true$$

Algorithm 3 is a formal representation of the resolution process. The DeLP program $\mathcal{P}$ is asked to produce an argumentation for each of the four crops to resolve the classification debate. If exactly one crop argumentation is successful then this crop is resolving the classification. Otherwise, the classification is left undecided. The answers to the four queries, corresponding to the four crops, are stored in the vector: $answer = \langle answer_{corn}, answer_{rice}, answer_{cotton}, answer_{soybean}\rangle$ where each element consists of a pair $\langle b, \mathcal{T}\rangle$, where b can be *True* or *False* and $\mathcal{T}$ is the dialectical tree. The three possible configurations for an $answer_y$ pair are:

- $\langle True, \mathcal{T}\rangle$, if y is warranted
- $\langle False, \mathcal{T}\rangle$, if $\sim y$ is warranted

Algorithm 3: DelpResolution : producing the resolved crop class by DeLP

Input: $\mathcal{P}$, a DeLP program
Input: $\mathcal{Y} = \{corn, cotton, soybean, rice\}$
Output: $y \in \mathcal{Y}$, the resolved crop class
Output: $\mathcal{T}$, dialectical tree

```
answer ← ⟨null, null, null, null⟩
foreach class y ∈ 𝒴 do
    𝒯 ← Build dialectical tree to warrant y over 𝒫
    if root(𝒯) is labeled Undefeated then
        answer_y ← ⟨True, 𝒯⟩
    else
        𝒯̄ ← Build dialectical tree to warrant ∼y over 𝒫
        if root(𝒯̄) is labeled Undefeated then
            answer_y ← ⟨False, 𝒯̄⟩
        else
            answer_y ← ⟨False, null⟩
        end
    end
end
if ∃!y ∈ 𝒴 s.t. answer_y=True and ∀ z ≠ y answer_z ≠ True then
    return y, 𝒯
else
    return undecided, 𝒯
end
```

- $\langle False, null \rangle$, if nor y neither $\sim y$ are warranted

In line with DeLP interpreter answers [47], $\langle True, \mathcal{T} \rangle$ corresponds to YES, $\langle False, \mathcal{T} \rangle$ to NO and $\langle False, null \rangle$ to UNDECIDED. The fourth possible status UNKNOWN is ignored as it can arise exclusively when y is not found in the program $\mathcal{P}$.

Within the conflict resolution algorithm 3, the *answer* vector is initialized on line 1. The pair $answer_y$ is built for each crop by a loop (lines 2-11). First, the algorithm tries to warrant y by building a dialectical tree having an *undefeated* labeled root (lines 3-5). If it succeeds, the answer is marked as positive. Contrary, it tries to warrant the complement $\sim y$ such that a negative answer can be inferred by a dialectical tree(lines 7-9). If nor y neither $\sim y$ are warranted, the answer is marked as negative (line 11). Once all the four answers are built, we check if there is exactly one that came up positive (line 12). If there is such an answer, then its crop class is considered the true class of the crop instance (line 13). Otherwise, the classification is left undecided (line 14).

Example 7 *Consider a debatable instance which produces the following answer vector within algorithm 3:* $answer = \langle answer_{corn}, answer_{rice}, answer_{cotton}, answer_{soybean} \rangle$*, where:*

$$\begin{aligned} answer_{corn} &= \langle False, \mathcal{T}_{corn} \rangle \\ answer_{rice} &= \langle False, \mathcal{T}_{rice} \rangle \\ answer_{cotton} &= \langle True, \mathcal{T}_{cotton} \rangle \\ answer_{cotton} &= \langle False, null \rangle \end{aligned}$$

The algorithm returns cotton because it corresponds to the exactly one True answer $answer_{cotton}$*. Both* $answer_{corn}$ *and* $answer_{rice}$ *produce False answers because their complement is warranted, producing dialectical trees having undefeated nodes as roots. The* $answer_{soybean}$ *can not produce any dialectical tree having an undefeated root, thus producing a False answer too.*

2.4.2 Dialectical analysis of a debatable instances

This section explains the dialectical analysis approach on one example of a debatable instance from the crop dataset.

Instance 32 of the crops dataset is classified as *cotton* by the decision tree classifier and *soybean* by the neural network and the support vector machine classifiers. By employing the voting resolution strategy, *soybean* would be declared the winning class with two votes against one. However, the actual class of the instance is *cotton*, correctly pointed by the described DeLP inference.

The feature values of the debatable instance are $g = 2.02, m = 1.85, ndvi = -1.16$. Expert knowledge is augmented with the phenological properties, *plant* on *May 10*, *harvest* on *Oct 15* with a true value for *harvest_color_change*. The rules extracted from the three statistical classifiers are identical with the ones in Examples 3, 4, and 5. The classifier rules are merged with the expert rules defined in Section 2.3.2 to form the DeLP program.

Finally, four queries, one for each crop type, are executed:

Corn query returns a False answer because the complement of *corn* is warranted. The argument structure $\langle \mathcal{A}_1, \sim corn \rangle$ is produced by the h_{dt} classifier, which believes that this instance should not be classified as corn:

$$\mathcal{A}_1 = \left\{ \begin{array}{l} \sim corn \mathrel{-\!\!\prec} decision\_tree(g, m, ndvi) \\ decision\_tree(g, m, ndvi) \mathrel{-\!\!\prec} m > -0.01, ndvi \in [-1.31, 0.07] \end{array} \right\}$$

$\langle \mathcal{A}_1, \sim corn \rangle$ is defeated by $\langle \mathcal{A}_2, corn \rangle$ and $\langle \mathcal{A}_3, corn \rangle$, argument structures produced by the expert rules confirming that plant date, harvest date and harvest state are specific for corn:

$$\mathcal{A}_2 = \left\{ \begin{array}{l} corn \mathrel{-\!\!\prec} expert\_corn(plant), expert\_corn(harvest) \\ expert\_corn(plant) \\ expert\_corn(harvest) \end{array} \right\},$$

$$\mathcal{A}_3 = \left\{ \begin{array}{l} corn \mathrel{-\!\!\prec} harvest\_color\_change \\ harvest\_color\_change \end{array} \right\}$$

$\langle \mathcal{A}_2, corn \rangle$ and $\langle \mathcal{A}_3, corn \rangle$ are in turn defeated by $\langle \mathcal{A}_4, \sim corn \rangle$, by the fact that green level does not fall in the expert defined range for corn:

$$\mathcal{A}_4 = \left\{ \begin{array}{l} \sim corn \mathrel{-\!\!\prec} \ \sim expert\_corn(g) \\ \sim expert\_corn(g) \end{array} \right\}$$

There is no other argument that can be constructed to defeat $\langle \mathcal{A}_4, \sim corn \rangle$, thus $\langle \mathcal{A}_1, \sim corn \rangle$ is reinstated. The dialectical tree that warranted $\sim corn$ is:

$$\begin{array}{ccc} & \langle \mathcal{A}_1, \sim corn \rangle^U & \\ \swarrow & & \searrow \\ \langle \mathcal{A}_2, corn \rangle^D & & \langle \mathcal{A}_3, corn \rangle^D \\ | & & | \\ \langle \mathcal{A}_4, \sim corn \rangle^U & & \langle \mathcal{A}_4, \sim corn \rangle^U \end{array}$$

Rice query returns a False answer. The complement of *rice* is warranted by a sole argument structure $\langle \mathcal{A}_5, \sim rice \rangle$, produced by the strict fact that rice is not changing significantly the color at harvest.

$$\mathcal{A}_5 = \left\{ \begin{array}{l} \sim rice \leftarrow harvest\_color\_change \\ harvest\_color\_change \end{array} \right\},$$

The corresponding dialectical tree is formed by a single node:

$$\langle \mathcal{A}_5, \sim rice \rangle^U$$

Cotton query produces a True answer because *cotton* is warranted. The argument structure $\langle \mathcal{A}_6, cotton \rangle$ is produced by the h_{dt} which believes that this instance should classified as cotton:

$$\mathcal{A}_6 = \left\{ \begin{array}{l} cotton \prec decision\_tree(g, m, ndvi) \\ decision\_tree(g, m, ndvi) \prec m > -0.01, ndvi \in]-1.31, 0.07] \end{array} \right\}$$

The h_{nn} and h_{svm} classifiers argue that the instance should be not classified as cotton, producing argument structures $\langle \mathcal{A}_7, \sim cotton \rangle$ and $\langle \mathcal{A}_8, \sim cotton \rangle$ that defeat the initial argument structure $\langle \mathcal{A}_6, cotton \rangle$:

$$\mathcal{A}_7 = \left\{ \begin{array}{l} \sim cotton \prec neural\_net(g, m, ndvi) \\ neural\_net(g, m, ndvi) \prec 0.21 * m - 1.44 * ndvi > 1.83 \end{array} \right\},$$

$$\mathcal{A}_8 = \left\{ \begin{array}{l} \sim cotton \prec svm(g, m, ndvi) \\ svm(g, m, ndvi) \prec g > 0.001, ndvi \leq -1.1 \end{array} \right\}$$

$\langle \mathcal{A}_9, cotton \rangle$ argument structure is produced by the expert rules and it defeats the h_{nn} and h_{svm} arguments $\langle \mathcal{A}_7, \sim cotton \rangle$ and $\langle \mathcal{A}_8, \sim cotton \rangle$. $\langle \mathcal{A}_9, cotton \rangle$ is derived from the fact that plant and harvest dates fit in the dates defined by the expert for planting and harvesting cotton:

$$\mathcal{A}_9 = \left\{ \begin{array}{l} cotton \prec expert\_cotton(plant), expert\_cotton(harvest) \\ expert\_cotton(plant) \\ expert\_cotton(harvest) \end{array} \right\},$$

Since there are no defeaters for $\langle \mathcal{A}_9, cotton \rangle$ the dialectical inference stops. The corresponding dialectical tree is:

$$\begin{array}{ccc} & \langle \mathcal{A}_6, cotton \rangle^U & \\ \langle \mathcal{A}_7, \sim cotton \rangle^D & & \langle \mathcal{A}_8, \sim cotton \rangle^D \\ | & & | \\ \langle \mathcal{A}_9, cotton \rangle^U & & \langle \mathcal{A}_9, cotton \rangle^U \end{array}$$

Soybean query produces a False answer because neither *soybean* nor $\sim$*soybean* are warranted. DeLP inference engine produces five dialectical trees all having the root labeled as *Defeated*. Since dialectical analysis can not prove *soybean* as the class of the instance the query returns *False*.

Two of the *soybean* dialectical trees have a similar inference process, differing only by the argument structures corresponding to the root nodes:

- $\langle \mathcal{A}_{10}, soybean \rangle^D$
  - $\langle \mathcal{A}_{11}, \sim soybean \rangle^U$
    - $\langle \mathcal{A}_{12}, soybean \rangle^D$
      - $\langle \mathcal{A}_{14}, \sim soybean \rangle^U$
    - $\langle \mathcal{A}_{13}, soybean \rangle^D$
      - $\langle \mathcal{A}_{14}, \sim soybean \rangle^U$

- $\langle \mathcal{A}_{15}, soybean \rangle^D$
  - $\langle \mathcal{A}_{11}, \sim soybean \rangle^U$
    - $\langle \mathcal{A}_{12}, soybean \rangle^D$
      - $\langle \mathcal{A}_{14}, \sim soybean \rangle^U$
    - $\langle \mathcal{A}_{13}, soybean \rangle^D$
      - $\langle \mathcal{A}_{14}, \sim soybean \rangle^U$

The root node arguments $\langle \mathcal{A}_{10}, soybean \rangle$ and $\langle \mathcal{A}_{15}, soybean \rangle$ are an outcome of the h_{svm} and h_{nn} learners, which both believe the instance should be classified as soybean:

$$\mathcal{A}_{10} = \left\{ \begin{array}{l} soybean \prec svm(g, m, ndvi) \\ svm(g, m, ndvi) \prec g > 0.001, ndvi \leq -1.1 \end{array} \right\}$$

,

$$\mathcal{A}_{15} = \left\{ \begin{array}{l} soybean \prec neural\_net(g, m, ndvi) \\ neural\_net(g, m, ndvi) \prec 0.21 * m - 1.44 * ndvi > 1.83 \end{array} \right\}$$

$\langle \mathcal{A}_{10}, soybean \rangle$ and $\langle \mathcal{A}_{15}, soybean \rangle$ are disputed by the h_{dt} classifier, which believes that the instance should not be classified as soybean, based on the argument $\langle \mathcal{A}_{11}, \sim soybean \rangle$:

$$\mathcal{A}_{11} = \left\{ \begin{array}{l} \sim soybean \prec decision\_tree(g, m, ndvi) \\ decision\_tree(g, m, ndvi) \prec m > -0.01, ndvi \in] -1.31, 0.07] \end{array} \right\}$$

The decision tree argument structure $\langle \mathcal{A}_{11}, \sim soybean \rangle$ is defeated by the expert derived arguments $\langle \mathcal{A}_{12}, soybean \rangle$ and $\langle \mathcal{A}_{13}, soybean \rangle$, which state that the expert green, moisture, ndvi levels and harvesting color state indicate soybean:

$$\mathcal{A}_{12} = \left\{ \begin{array}{l} soybean \prec expert\_soybean(g), expert\_soybean(m), expert\_soybean(ndvi) \\ expert\_soybean(g) \\ expert\_soybean(m) \\ expert\_soybean(ndvi) \end{array} \right\},$$

$$\mathcal{A}_{13} = \left\{ \begin{array}{l} soybean \prec harvest\_color\_change \\ harvest\_color\_change \end{array} \right\}$$

$\langle \mathcal{A}_{12}, soybean \rangle$ and $\langle \mathcal{A}_{13}, soybean \rangle$ are in turn defeated by the expert argument structure $\langle \mathcal{A}_{14}, \sim soybean \rangle$, pointing that planting date is outside of *soybean* planting time-frame:

$$\mathcal{A}_{14} = \left\{ \begin{array}{l} \sim soybean \prec \sim expert\_soybean(plant) \\ \sim expert\_soybean(plant) \end{array} \right\}$$

There are no argument structures that can be posed to $\langle \mathcal{A}_{14}, \sim soybean \rangle$, thus the argument is undefeated. The dialectical analysis ends for the two trees which started with arguments from the h_{svm} and h_{nn} classifiers.

Two more soybean dialectical trees with *Defeated* root nodes are produced based on expert argument structures $\langle \mathcal{A}_{12}, soybean \rangle$ and $\langle \mathcal{A}_{14}, \sim soybean \rangle$. The first argument states that soybean is a possible match because green, moisture, and ndvi levels correspond to soybean. The second argument opposes to soybean because the planting date does not fall in the expert defined time-frame. The two dialectical trees are:

$$\begin{array}{c} \langle \mathcal{A}_{12}, soybean \rangle^D \\ | \\ \langle \mathcal{A}_{14}, \sim soybean \rangle^U \end{array}$$

$$\begin{array}{c} \langle \mathcal{A}_{14}, \sim soybean \rangle^D \\ | \\ \langle \mathcal{A}_{12}, soybean \rangle^U \end{array}$$

The last soybean dialectical tree fails to warrant $\sim soybean$ by starting with the argument structure induced by the decision tree learner $\langle \mathcal{A}_{11}, \sim soybean \rangle$. The h_{svm} classifier defeats $\langle \mathcal{A}_{11}, \sim soybean \rangle$ by using its argument $\langle \mathcal{A}_{10}, soybean \rangle$. The argument conveyed by h_{svm} is in turn defeated by the expert using $\langle \mathcal{A}_{14}, \sim soybean \rangle$ arguing that planting date is outside of soybean planting period. Finally $\langle \mathcal{A}_{14}, \sim soybean \rangle$ is defeated by $\langle \mathcal{A}_{12}, soybean \rangle$ expert argument which says that green, moisture and ndvi levels are specific for soybean. The dialectical tree is:

$$\begin{array}{c} \langle \mathcal{A}_{11}, \sim soybean \rangle^D \\ | \\ \langle \mathcal{A}_{10}, soybean \rangle^U \\ | \\ \langle \mathcal{A}_{14}, \sim soybean \rangle^D \\ | \\ \langle \mathcal{A}_{12}, soybean \rangle^U \end{array}$$

Table 2.3: Conflict resolution accuracy, precision and recall on the conflict set Γ of 306 instances.

Method	Accuracy	Corn		Rice		Cotton		Soybean	
		P	R	P	R	P	R	P	R
Voting	58.1	45	75	90.6	72.5	57.6	70	50	36.8
DeLP	99	100	75	100	100	100	100	100	100

Table 2.4: Confusion matrix for DeLP resolution classifier on the conflict set Γ.

	Corn	Rice	Cotton	Soybean	Undecided	Recall
Corn	9	0	0	0	3	75
Rice	0	40	0	0	0	100
Cotton	0	0	140	0	0	100
Soybean	0	0	0	114	0	100
Precision	100	100	100	100		

Out of the four queries: *corn*, *cotton*, *rice* and *soybean*, the only *True* answer is outputted by the *cotton* query, which corresponds to the actual class.

2.4.3 Experimental results

The results are presented first from the conflict resolution perspective, evaluating the resolution methods over the set of conflicting instances. Then, classification results are presented for the entire test crops dataset.

The set of conflicting instances is formed by 306 cases in which the statistical classifiers gave conflicting predictions. The conflict set accounts for 7% of the test dataset. Table 2.3 lists the results of conflict resolution methods employed on the conflict set. Resolving conflicts by voting gave an accuracy of 58.1%. On the other hand, resolving conflicts by DeLP argumentation, making use of expert and base classifiers knowledge, a much higher accuracy of 99% was obtained. Figure 2.6 displays the classification result for each pixel of the conflict set.

As the voting system does not use human knowledge, while our ensemble-delp resolutor uses rules extracted from the agricultural domain, the improvement from 58.1% to 99% accuracy on the conflict set quantifies the impact of human knowledge in the classification. Hence, the difference of 99%-58.1%=40.9% percents is due to the expert rules and the conflict resolution strategy of our argumentation method. This percent of 40.9% increasing represents the quantification of the relevance of the human knowledge in this domain.

(a) Argumentation conflict resolution (99% accuracy).

(b) Voting conflict resolution (58.1% accuracy)

Figure 2.6: Classification results on conflict set Γ. The color codes for *corn*, *rice*, *cotton* and *soybean* are light blue, blue, light green and green respectively.

DeLP conflict resolution produced a precision of 100% for each crop class, all predicted values being correctly classified. DeLP left three conflicting instances un-classified thus not achieving a perfect recall for *corn* (75%) class. Table 2.4 presents the confusion matrix for DeLP classification resolution on the conflict set. Three *corn* instances remained unclassified because DeLP inferred that none of the four classes is a match for these instances.

Voting resolution method produced lower precision scores on the conflict set, especially for *corn* (45%), *soybean* (50%), and *cotton* (57.6%). Table 2.5 presents the confusion matrix for voting classification resolution on the conflict set. The low precision for *corn* is caused by incorrectly classifying more than half *corn* instances as *rice*. The low precision for *soybean* is caused by incorrectly classifying half of the *soybean* instances as *cotton*. DeLP resolution was able to settle all these confusions by making use of expert knowledge. For example *corn* is differentiated

Table 2.5: Confusion matrix for voting resolution classifier on the conflict set Γ.

	Corn	Rice	Cotton	Soybean	Recall
Corn	9	3	0	0	75
Rice	11	29	0	0	72.5
Cotton	0	0	98	42	70
Soybean	0	0	72	42	36.8
Precision	45	90.6	57.6	50	

Table 2.6: Classification accuracy, precision and recall per each classification method on the test dataset (4,342 instances).

Classification Method	Acc.	Corn		Rice		Cotton		Soybean	
		P	R	P	R	P	R	P	R
Ensemble Voting	95.5	99.5	99.8	95.1	80.8	84.9	93.1	89.1	77.4
Ensemble DeLP	**98.4**	**99.8**	**99.9**	**100**	**95.8**	**93.3**	**98.6**	**98**	**90**

from *rice* by the significant change in color when harvested, while planting season for *soybean* can overpass with one month the *cotton* planting season.

Table 2.6 lists the evaluation of all classification methods on the test dataset (4,342 instances), best results are shown in bold face. Due to better conflict resolution, the Ensemble using DeLP produced a higher accuracy (98.4%) than the ensemble using voting (95.5%).

McNemar's test is employed for showing the statistical significance of the classification methods. [45] advocates for using the McNemar's test for remote sensing to compare classifiers built by using the same dataset. To compare the performance between two classification methods, a value z is computed according to the formula: $z = \frac{f_{12} - f_{21}}{\sqrt{f_{12} + f_{21}}}$, where f_{12} represents the count of instances correctly classified by the first classifier and wrongly classified by the second, while f_{21} represents the count of instances correctly classified by the second classifier and wrongly classified by the first. According to [45], when $\mid z \mid > 1.96$ there is a difference in accuracy at a confidence level of 95%. For evaluating our classifiers, we computed the z score for the ensemble using argumentation and the ensemble using voting resolution. The z value was determined to be 11.1 (since $f_{12} = 125$ and $f_{21} = 0$), indicating a positive significance and thus a superior accuracy of the argumentation over the voting conflict resolution.

2.5 Conclusions

We developed a solution for conflict resolution is ensemble learning, and we successfully apply this solution for crop classification in the agriculture domain. Our hybrid system merges machine learning and symbolic argumentation with the scope of improving the classification of four crop classes in remote sensing: corn, soybean, rice and cotton. The machine learning pursuit is represented by an ensemble learner composed of three discriminative models: decision tree, neural network and support vector machine. Conflicting situations, characterized by instances for which base classifiers do not reach consensus, are resolved by using a symbolic argumentation process. Within the argumentation process, a dialectical analysis is performed on symbolic rules extracted from the base classifiers and knowledge defined by an expert. Expert knowledge guides the resolution process to reach definite decisions within a closed context defined from morphological and phenological profiles of the four crops. The proposed solution improved both the accuracy of resolution of conflicting instances and the accuracy of the ensemble learner as a whole. In conclusion, our argument-based conflict resolutor proved to be more effective than voting-based resolutor in ensemble learning. Moreover, the experiments clearly indicated the high impact of expert knowledge on resolving debatable classes in the agriculture domain.

The presented approach has several contributions in regards to the field of Expert and Intelligent Systems. To the best of our knowledge, this is the first approach that combines ensemble learning and argumentation in the agricultural domain. We developed a method for extracting defeasible rules from base learners to facilitate the integration of expert rules in the decision process. Moreover, the advantages the argumentation machinery brought on top of an ensemble classifier are: First, arguments helped us to introduce and use human knowledge during classification. Second, our experiments proved that argumentative reasoning represents a means to conflict resolution in ensemble learning, instead of voting-based methods. Third, by combining arguments with machine learning we managed to handle different types of information in a uniform way. Forth, argumentation increased transparency on our hybrid intelligent system. Hence, we consider that the conceptual instrumentation presented in this work can be used to take decisions in domains characterized by high data availability, robust expert knowledge, and a need for justifying the rationale behind decisions.

We argue here that the proposed argument-based ensemble learning is close to the human cognitive model: Firstly, [142] has explained that seeking additional opinions before making a decision is an innate behavior for human agents. Similarly, ensemble learning considers classification decisions from different base learners. Secondly, argumentative-based decisions often occur in daily human tasks instead of various algebraic-based methods for opinion aggregation. Similarly, rule-based argumentation performs dialectical reasoning to decide on a winning argument. The above two observations suggest that combining ensemble learning and argumentation fits the decision patterns of human agents, both in terms of collecting opinions and dialectical reasoning on these opinions. Moreover, both ensemble learning and argumentative-based reasoning help us to minimize the risk of taken an obviously wrong decision. First, ensemble learning diminishes the risk to rely on a single inadequate base classifier. Second, by providing the dialectical tree, defeasible rule-based argumentation helps the human agent to identify reasoning flaws of the rationale behind the decision.

In the context of the AI shifting towards machine learning and the bring some light to the machine learning black boxes [155] our hybrid intelligent system exploits both logic-based AI and statistical learning. In line with [155], our view is that knowledge representation can bring valuable benefits to the black boxes within most of the learning algorithms or probabilistic-based computations. We are aware of the difficulties of knowledge-based approaches to include new experiences, or to deal with non-linearity [9].

The solution proposed in this paper for crop classification lays the groundwork for several extensions: (1) using a more expressive argumentation model, such as weighted argumentation systems [39] or probabilistic argumentation [77]; (2) exploiting the available formal knowledge in the crop domain, by importing various agricultural ontologies in the expert knowledge base [25, 98, 16]; (3) investigating the behavior of the system in case of large numbers of base learners, towards large scale argumentation on crowds of learners [99].

Chapter 3

Arguing in description logics

> "Knowledge is irreversible: one cannot go back into the darkness of sweet ignorance"
>
> Stanislaw Lem

Computational models of arguments are more realistic when they include concepts of both argumentation and explanation, as shown in the informal logic literature. Apart from distinguishing explanations from arguments, we present our approach for modeling them together. To describe the communicative acts of the agents, we consider their different views on the topics of the dialog. With the subjective views of the agents, we model the speech acts to enable the distinction between argument and explanation in utterances. By using description logics (DL) to define the ontologies of the agents, the DL reasoning tasks are used to distinguish an argument from an explanation. This chapter is an extension of [103].

3.1 Distinguishing argument from explanation

Argument and explanation are considered distinct and equally fundamental [126], with a complementary relationship [127], as a central issue for identifying the structure of natural dialogs. While argumentation brings practical benefits in persuasion, deliberation, negotiation, collaborative decisions, or learning [130],[153],[5], it also involves costs [140].

The problem addressed in this study is that of distinguishing between argument and explanation in natural dialog. Even if interleaving argument and explanation is common practice in daily communication, the task of extending argumentation theory with the concept of explanation is still at the very early stages [173].

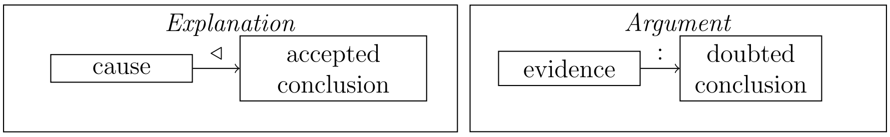


Figure 3.1: Distinguishing argument from explanation.

The fusion of argument and explanation is best shown by the fact that humans tend to make decisions both on knowledge and understanding [179]. For instance, in judicial cases, circumstantial evidence needs to be complemented by a motive explaining the crime, but the explanation itself is not enough without plausible evidence [127]. In both situations the pleading is considered incomplete if either argumentation or explanation is missing. Thus, the interaction between argument and explanation, known as *argument-explanation pattern*, has been recognized as the basic mechanism for augmenting an agent's knowledge and understanding [35].

The role of argument is to establish knowledge, while the role of explanation is to facilitate understanding [127]. To make an instrumental distinction between them, one has to distinguish between knowledge and understanding. We consider the following distinctive features of argument and explanation:

- *Starting condition.* Explanation starts with non-understanding. Argumentation starts with a conflict.
- *Role symmetry.* In explanation the roles are usually asymmetric: the explainer is assumed to have more understanding and aims to transfer it to the explainee. In argumentation, both parties start the debate from equal positions, thus initially having the same roles. Only at the end of the debate the asymmetry arises when the winner is considered to have more relevant knowledge on the subject. If no winner occurs, the initial symmetry between arguers is preserved.
- *Linguistic indicator.* In explanation one party supplies information. There is a linguistic indicator which requests that information. Because in argumentation it is assumed that all parties supply information, no indicator of demanding the information is required.
- *Acceptance.* An argument is accepted or not, while an explanation may have levels of acceptance.

Regarding the "starting condition", for an argument, premises represent evidence supporting a doubted conclusion. For an explanation, the conclusion is accepted and the premises represent the causes of the consequent (see Fig. 3.1). The

Table 3.1: Explanations versus arguments.

	Explanation	Argument
Consequent	Accepted as a fact	Disputed by parties
Premises	Represent causes	Represent evidence
Reasoning Pattern	Provides less well known statements why a better known statement is true	From well known statements to statements less well known
Answer to	Why is that so?	How do you know?
Contribute to	Understanding	Knowledge
Acceptance	Levels of understanding	Yes/No

explanation aims to understanding the explanandum by indicating what causes it, while an argument aims to persuade the other party about a believed state of the world. An argument is considered adequate in principle if there is at least one agent who justifiably believes that the premises are true but who does not justifiably believe this about the consequent [120]. An explanation is adequate in principle if all the agents accepting the premises would also accept the consequent. The function of argument is to "transfer a justified belief", while the role of explanation is to "transmit understanding". Therefore, unlike arguments, the statements in an explanation link well known consequents to less known premises [81].

Regarding the "role symmetry", consider the dialog between a teacher and a junior student which is almost entirely explicative. The ontology of the student regarding the specific scientific field is included in the ontology of the teacher. As the ontology of the student increases, resulting in different perspectives on the subject, exchanging arguments may occur.

For "linguistic indicator", the easiest way to distinguish between explanation and argument is to compare arguments *for* F and explanations *of* F. The mechanism should distinguish between whether F is true and why F is true. In case F is a normative sentence, the distinction is difficult [179]. If F is an event, the question why F happened is clearly delimited by whether F happened.

The "acceptance" topic is supported by the fact that, unlike knowledge, understanding admits degrees [88]. The smallest degree of understanding, making sense, demands a coherent explanation, which usually is also an incomplete one. It means that, when the explainer conveys an "I understand" speech act, the explainer can shift to an examination dialog in order to figure out the level of understanding, rather than a crisp value understand/not understand, as investigated by Walton [170]. Acceptability standards for evaluating explanation can be defined similarly to standards of proof in argumentative theory [52]. Some elements that help to distinguish between argument and explanation are shown in the Table 3.1.

3.2 Formalising argument and explanation

The first part of this section models the distinguishing features of arguments and explanation in the description logic. The second part of this section uses rules on top of DL to model the reasoning patterns in which both an argument and an explanation supports a claim.

3.2.1 Arguments and explanations in description logic

At the top level of our argument and explanation ontology (*ArgExp*), we have *statements* and *reasons*. A statement claims a text of type string, given by: $Statement \sqsubseteq \exists\, claimsText.String$.

Definition 5 *A reason consists of a set of premises supporting one conclusion.*

$$Reason \sqsubseteq \exists\, hasPremise.Statement \sqcap (=1) hasConclusion.Statement \tag{3.1}$$

Arguments and explanations are forms of reasoning.

Definition 6 *An argument is a reason in which the premises represent evidence in support of a doubted conclusion.*

$$Argument \sqsubseteq Reason \sqcap \forall\, hasPremise.Evidence \sqcap (=1) hasConclusion.DoubtedSt \tag{3.2}$$

Definition 7 *An explanation is a reason in which the premises represent a cause of an accepted fact.*

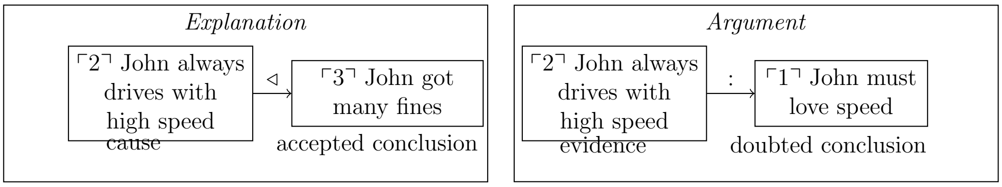
$$Explanation \sqsubseteq Reason \sqcap \forall\, hasPremise.Cause \sqcap (=1) hasConclusion.Fact \tag{3.3}$$

We define a doubted statement as a statement attacked by another statement:

$$DoubtedSt \equiv Statement \sqcap \exists\, attackedBy.Statement \tag{3.4}$$

The domain of the role *attackedBy* is a *Statement*, formalised by $\exists\, attackedBy.\top \sqsubseteq Statement$, while its range is the same concept *Statement*: $\top \sqsubseteq \forall\, attackedBy.Statement$. The role *attacked* is the inverse role for *attackedBy*, expressed in DL with $attack^{-} \equiv attackedBy$.

A fact is a statement which is not doubted: $Fact \equiv Statement \sqcap \neg DoubtedSt$. Note that facts and doubted statements are disjoint ($Fact \sqcap DoubtedStatement \sqsubseteq \bot$). Pieces of evidence and cause represent statements: $Evidence \sqsubseteq Statement$, $Cause \sqsubseteq Statement$. The concepts for evidence and cause are not disjoint: the same sentence can be interpreted as evidence in one reason and as cause in another reason, as illustrated in Example 8.



$\ulcorner 2 \urcorner : \mathit{Cause}$ $\quad$ $\ulcorner 3 \urcorner : \mathit{Fact}$ $\quad$ $\ulcorner 2 \urcorner : \mathit{Evidence}$ $\quad$ $\ulcorner 1 \urcorner : \mathit{DoubtedSt}$

$e : \mathit{Reason}$ $\quad$ $a : \mathit{Reason}$

$(e, \ulcorner 2 \urcorner) : \mathit{hasPremise}$ $\quad$ $(e, \ulcorner 3 \urcorner) : \mathit{hasConclusion}$ $\quad$ $(a, \ulcorner 2 \urcorner) : \mathit{hasPremise}$ $\quad$ $(a, \ulcorner 1 \urcorner) : \mathit{hasConclusion}$

Figure 3.2: The same statement $\ulcorner 2 \urcorner$ acts as a cause for the accepted statement $\ulcorner 3 \urcorner$ and as evidence for doubted statement $\ulcorner 1 \urcorner$. The agent with this interpretation function treats e as an explanation (e : *Explanation*) and a as an argument (a : *Argument*).

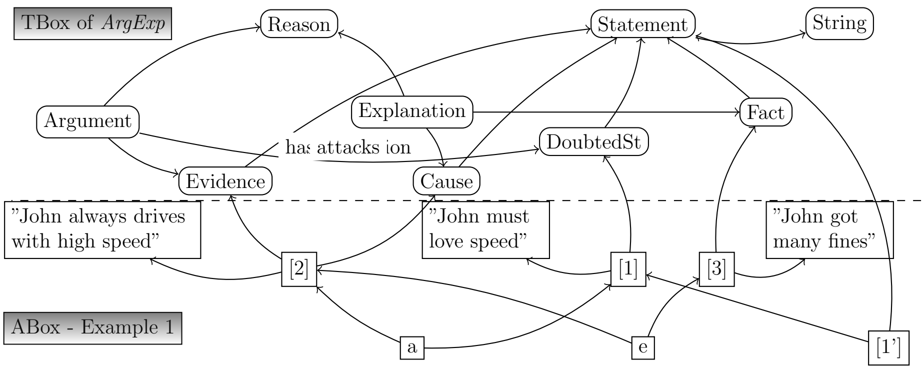


Figure 3.3: Vizualising the Tbox and Abox of the agent h in Example 8.

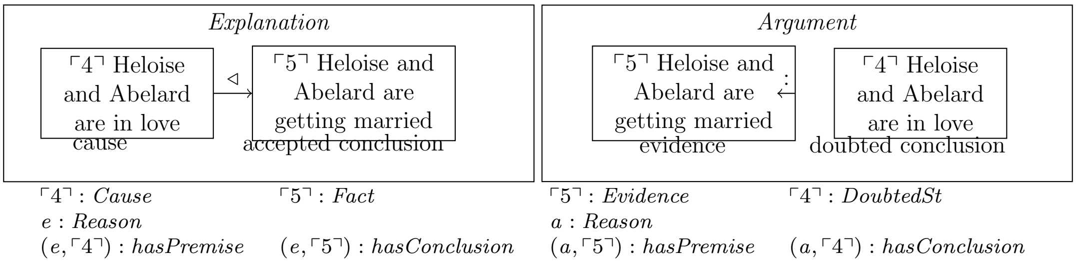


Figure 3.4: Opposite interpretations of the same reason: left: e is classified as an explanation; right - a is interpreted as an argument.

Example 8 (Different interpretations of the same premise.) *Consider the following statements:*

John must love speed.	$\ulcorner 1 \urcorner$
He drives with high speed all the time.	$\ulcorner 2 \urcorner$
That's why he got so many fines.	$\ulcorner 3 \urcorner$

One possible interpretation is that statement $\ulcorner 2 \urcorner$ *represents the support for statement* $\ulcorner 1 \urcorner$. *Statement* $\ulcorner 2 \urcorner$ *also acts as an explanation for* $\ulcorner 3 \urcorner$, *as suggested by the textual indicator "That's why". Fig. 3.2 illustrates the fomalisation in DL of these two reasons. Assume that the interpretation function* $\mathfrak{I}$ *of the hearing agent* h *asserts statement* $\ulcorner 2 \urcorner$ *as an instance of the concept Cause and* $\ulcorner 3 \urcorner$ *as a Fact. Based on axiom 3.3, agent* h *classifies the reason* e *as an explanation.*

Let the Abox of agent h *contains also the assertion* ($\ulcorner 1' \urcorner$,$\ulcorner 1 \urcorner$)*:attacks. Based on axiom 3.4, agent* h *classifies the statement* $\ulcorner 1 \urcorner$) *as doubted. Adding that* $\ulcorner 2 \urcorner$) *is interpreted as evidence, agent* h *classifies the reason* a, *based on definion 3.2. The relations among individuals in the Example 8 are depicted in the bottom of the Fig. 3.3, where the top level concepts of our argument-explanation ontology ArgExp are also illustrated. Based on the definitions in the TBox and the instances of the ABox,* a *is an argument and* e *is an explanation.*

In agent communication, agents can have different interpretation functions of the same chain of conveyed statements. In Example 9, the agents have opposite interpretation regarding the premise and the conclusion of the same reason.

Example 9 (Opposite interpretations of the same reason.) *Consider the following reason containing the statements* $\ulcorner 4 \urcorner$ *and* $\ulcorner 5 \urcorner$*:*

Heloise and Abelard are in love.	$\ulcorner 4 \urcorner$
Heloise and Abelard are getting married.	$\ulcorner 5 \urcorner$

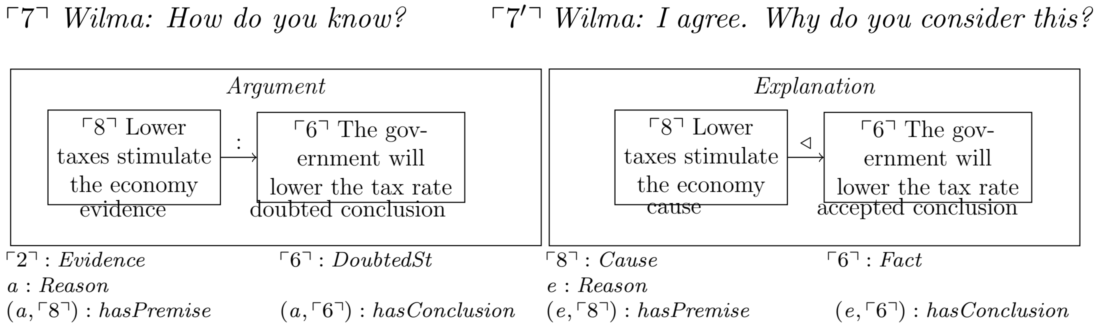


Figure 3.5: The dialog provides indicators helping Bob to assess the status of the consequent from Wilma's perspective: In the left part, query $\ulcorner 7 \urcorner$ does not suggest the acceptance of conclusion $\ulcorner 6 \urcorner$. In the right part, answer $\ulcorner 7' \urcorner$ clearly indicates the Wilma also accepts claim $\ulcorner 6 \urcorner$.

The ambiguity arises from the difficulty to identify which is the premise and which the conclusion. One agent can interpret $\ulcorner 4 \urcorner$ *as a cause for the accepted fact* $\ulcorner 5 \urcorner$*, treating the reason* e *as an explanation (left part of Fig. 3.4). Here,* $\ulcorner 4 \urcorner$ *acts as a premise in the first interpretation (left part) and as a conclusion in the second one (right part). An agent with a different interpretation function* $\mathfrak{I}$ *will assert* $\ulcorner 5 \urcorner$ *as evidence for the doubted conclusion* $\ulcorner 4 \urcorner$*, therefore rising an argument.*

How can the agents exploit the information that the given dialog is interpreted as an explanation by one party and as an argument by the other, in order to eliminate the ambiguity. Consider the following dialog adapted from [21]:

Bob:	*The government will inevitably lower the tax rate.*	$\ulcorner 6 \urcorner$
Wilma:	*How do you know?*	$\ulcorner 7 \urcorner$
Bob:	*Because lower taxes stimulate the economy.*	$\ulcorner 8 \urcorner$

The dialog is shown in the Fig. 3.5 as an argument with the consequent $\ulcorner 6 \urcorner$ supported by the premise $\ulcorner 8 \urcorner$. Let's assume that Wilma's reply is slightly modified, given by: $\ulcorner 7' \urcorner$ Wilma: *I agree. Why do you consider this?*

By accepting statement $\ulcorner 6 \urcorner$, it becomes a fact in the situation represented by Bob and Wilma. Consequently, the reason becomes an explanation in which the cause "lower taxes stimulate the economy“ may explain the government's decision (Fig. 3.5). Assuming that an agent accepts a statement only if it has a level of understanding of that sentence, one can infer that Wilma has her own explanation regarding the fact $\ulcorner 6 \urcorner$, but she wants to find out her partner's explanation.

Given the difficulty to distinguish between causes and evidence, a simplified argument-explanation model would consider only the status of the consequent. Thus, if an agent accepts the conclusion according to its interpretation function,

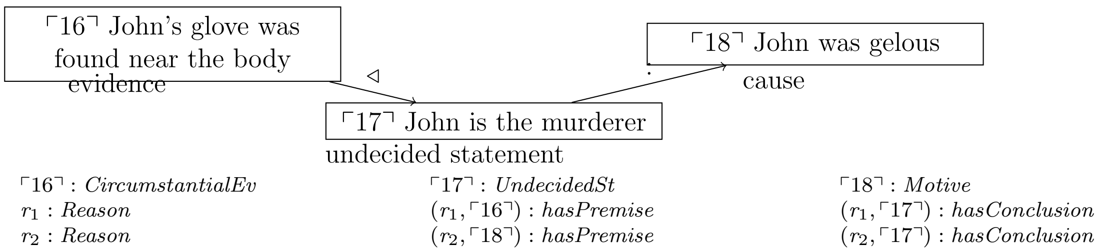


Figure 3.6: Argument-explanation pattern supporting consequent.

Agent A ($\mathcal{O}_A$)
$u : GoodUniversity$
$GoodUniversity \sqsubseteq \exists\, hasGood.ResearchFacility$

Agent's A view on agent B ($\mathcal{O}_{AB}$)
$u : GoodUniversity$
$GoodUniversity \equiv \exists\, hasGood.ResearchFacility \sqcap \exists\, hasGood.TeachingFacility$

Agent B view on agent A ($\mathcal{O}_{BA}$)
$u : ResearchInstitute$
$ResearchInstitute \sqsubseteq \exists\, hasGood.ResearchFacility$

Agent B ($\mathcal{O}_B$)
$u : GoodUniversity$
$GoodUniversity \equiv \exists\, hasGood.ResearchFacility \sqcup \forall\, hasGood.TeachingFacility$

Figure 3.7: Subjective views of agents.

then it treats the premise as cause (axiom 3.6). If the agent interpretes the conclusion as doubted, it will treat the premise as evidence (axiom 3.6).

$$\exists\, hasPremise^{-}.(Reason \sqcap \exists\, hasConclusion.Fact) \sqsubseteq Cause \qquad (3.5)$$

$$\exists\, hasPremise^{-}.(Reason \sqcap \exists\, hasConclusion.DoubtedSt) \sqsubseteq Evidence \qquad (3.6)$$

3.3 The subjective views of the agents

The agents construct arguments and explanations from their own knowledge bases which do no completely overlap. At the same time, each party has a subjective model about the knowledge of its partner.

Let's consider the partial knowledge in the Fig. 3.7. Here the agent A sees the individual u as a good university, where a good university is something included in all objects for which the role *hasGood* points towards concepts of type *ResearchFacility*. According to the agent B ontology ($\mathcal{O}_B$), u is also a good university, but the definition is more relaxed: something is a good university if it has at least one good research facility or all the teaching facilities are good.

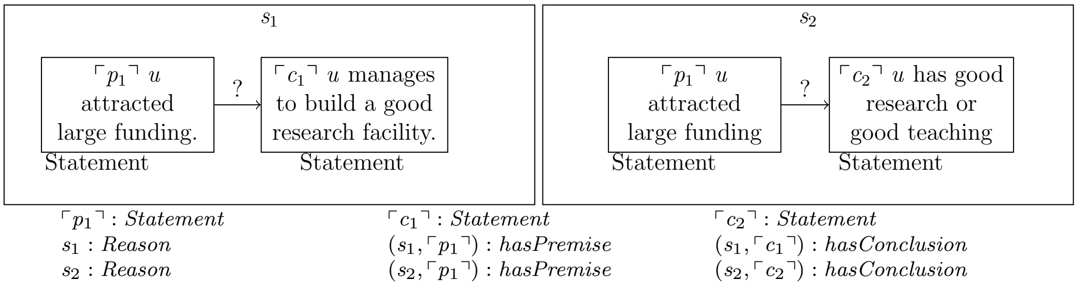


Figure 3.8: Reasons conveyed by the agent A. Are they arguments or explanations?

According to the agent A's perspective on the knowledge of the agent B ($\mathcal{O}_{AB}$), u belongs to the concept of good universities, but the definition is perceived as being more restrictive: a good university should have at least one good research facility but also at least one good teaching facility. From the opposite side ($\mathcal{O}_{BA}$), the agent B perceives that A asserts u as a research institute, where a research institute should have good research facility.

Suppose that the agent A conveys different reasons s_1 and s_2 supporting the statement c_1: "u has good research facility" and c_2: "u has either good research or good teaching". For instance:

s_1: "Because u attracted large funding from research projects, u manages to build a good research facility."

s_2: "Because u attracted large funding from research projects, u should have either good research or good teaching."

The reasons s_1 and s_2 are graphically represented in the Fig. 3.8. Let's assume that both agents formalize statements c_1 and c_2 as follows:

c_1 : "$u : \exists\, hasGood.ResearchFacility$"

c_2 : "$u : \exists\, hasGood.(ResearchFacility \sqcup Teaching)$"

How does the agent A treat one reason, when conveying it to the agent B, as explanation or argument?. Given the models in the Fig. 3.7, how does the receiving agent B perceive the reason: an explanatory or an argumentative one?

To distinguish between explanation and argument, the most important issue regards the acceptance of the consequent. In the Table 3.2, $\oplus$ denotes that the ontology $\mathcal{O}_X$ entails the consequent c_j. The statement c_1 can be derived from the ontology $\mathcal{O}_A$ (Fig. 3.7). It cannot be inferred (noted with $\ominus$) by the agent B based on its ontology $\mathcal{O}_B$, because in its interpretation a university which has only good teaching facilities, but no good research facility, is also a good university (given by the disjunction in the definition of *GoodUniversity* in $\mathcal{O}_B$).

Instead, the statement c_2 fits the definition of good ontology in $\mathcal{O}_B$. Because the agent A accepts its first part *"u has good research"*, it should also consider c_2: *"u has good research or good teaching"* as valid. Similarly, agent A considers that

Table 3.2: Entailment of statements c_1 and c_2 in agent ontology

Agents ontologies	$\models c_1$?	$\models c_2$?
$\mathcal{O}_A$	$\oplus$	$\oplus$
$\mathcal{O}_{AB}$	$\oplus$	$\ominus$
$\mathcal{O}_B$	$\ominus$	$\oplus$
$\mathcal{O}_{BA}$	$\oplus$	$\oplus$

Table 3.3: Acceptance of consequents c_1 and c_2 based on ontology

World	Ontologies	c_2	c_1
w_O	$\mathcal{O}_A + \mathcal{O}_B$	Accepted	Doubted
w_A	$\mathcal{O}_A + \mathcal{O}_{AB}$	Doubted	Accepted
w_B	$\mathcal{O}_B + \mathcal{O}_{BA}$	Accepted	Doubted

agent B cannot infer c_2 ($\ominus$ in Table 3.2), even if the $\mathcal{O}_B$ ontology entails c_2.

The agent A has a wrong representation $\mathcal{O}_{AB}$ regarding how the agent B views the statement c_2. Even if the agent B has a wrong model $\mathcal{O}_{BA}$, based on which it believes that the agent A interprets u as a research institute instead of a university, the consequent c_2 is still derived based on the axiom $\mathit{ResearchInstitute} \sqsubseteq \exists\, \mathit{hasGood}.\mathit{ResearchFacility}$.

The knowledge of the agent A ($\mathcal{O}_A$), and its model about the knowledge of B ($\mathcal{O}_{AB}$), represent the subjective world of the agent A, noted with w_A in the Table 3.3. Similarly, the subjective world w_B of the agent B consists of the knowledge of B ($\mathcal{O}_B$), and its view on the knowledge of the agent A. The knowledge of A combined with the knowledge of B ($\mathcal{O}_{BA}$), represent the objective world w_O. A statement is considered *Accepted* if it is entailed by both ontologies. If at least one ontology does not support the statement, it is considered *Doubted*. The following algebra encapsulates this:

$$\oplus + \oplus = \mathit{Accepted} \quad \oplus + \ominus = \mathit{Doubted}$$
$$\ominus + \oplus = \mathit{Doubted} \quad \ominus + \ominus = \mathit{Doubted}$$

In Table 3.3, agent A treats c_2 as accepted, meaning that from its point of view the reason s_2 represents an explanation. Agent B perceives the sentence c_2 as doubted, therefore it considers that it is hearing an argument (world w_A in the Fig. 3.9). Note that in the objective world w_O, the reason s_2 is actually an argument. That means that the agent A is wrong about the model of its partner

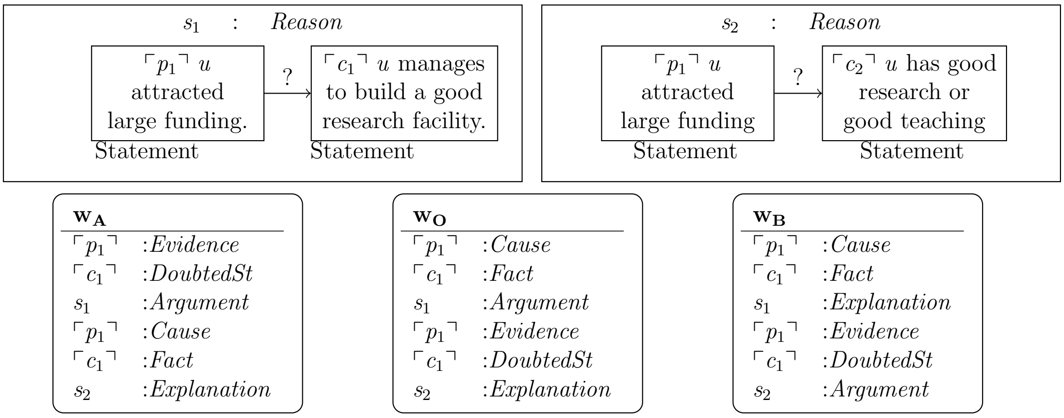


Figure 3.9: Interpreting reasons s_1 and s_2 in different worlds.

Table 3.4: Cases when X conveys/expects argument or explanation

	Communicate	Expects
Argument	$Doubted_X$	$\oplus_X^w \vee \ominus_X^{\neg w}$
Explanation	$Doubted_X$	$\ominus_X^w \vee \oplus_X^{\neg w}$

B. Consider that the reason s_1 is uttered by the agent B. It believes that it is conveying an argument, which is true in the objective world w_O. The agent A considers that it is receiving an explanation, which is false in w_O.

The statement c_1 being perceived as doubted in w_A, the agent A considers that it is conveying an argument. In the world w_B, the conclusion is accepted, thus the agent B is hearing an explanation, which is true in the objective world w_O. In this situation, the agent B should signal to its partner: *"There is no need to persuade me. I agree with the consequent."*

The correctness or adequacy of conveying either argument or explanation should be computed relative to the objective world w_O. Given the difference between expecting explanations or arguments (subjective worlds w_A and w_B) and legitimate ones (objective world w_O), the agents may wrongly expect explanations instead of arguments and vice versa. For the correctness or adequacy of conveying/expecting argument or explanation, the algebra in the Fig. 3.10 is used. The first operator represents the actual world w_O, while the second one represents the subjective perspective of the agent X.

By analyzing the entailment of a statement in all four knowledge bases, the situations in which the agents expect explanation or argument are synthesized in the Table 3.4. Assuming sincere agents, X conveys an argument if in its world the statement is *Doubted*. If the statement is *Accepted*, X conveys explanation. The agent X receives explanations when it is right about an agreement ($\oplus_X^w$) or when

$$
\begin{array}{ll}
Accepted_O + Accepted_X = \oplus_X^{w} & \text{agreement rightness}\\
Accepted_O + Doubted_X = \oplus_X^{\neg w} & \text{agreement not aware}\\
Doubted_O + Accepted_X = \ominus_X^{\neg w} & \text{conflict not aware}\\
Doubted_O + Doubted_X = \ominus_X^{w} & \text{conflict rightness}
\end{array}
$$

Figure 3.10: Correctness/inadvertence of expectation.

Table 3.5: Awareness regarding consequents c_1 and c_2.

Agent	Awareness and Ignorance	c_1	c_2
A	$w_O + w_A$	$\ominus_A^{\neg w}$	$\oplus_A^{\neg w}$
B	$w_O + w_B$	$\ominus_B^{w}$	$\oplus_B^{w}$

it is not aware of a conflict ($\ominus_X^{\neg w}$). It receives arguments either when X is aware of a disagreement ($\ominus_X^{w}$) or it is not aware of an agreement ($\oplus_X^{\neg w}$).

The situation resulting by applying the algebra in the Fig. 3.10 on the given scenario is presented in the Table 3.5. The agent B, even if its model about A is not accurate, manages to figure out the status of both consequents c_1 and c_2. Quite differently, the agent A is ignorant with respect to both conclusions.

Is it possible for the hearing agent to indicate to the conveyor agent that a wrong assumption has been made? The problem is that no agent is aware of the objective world w_O. The following two options may exist to handle this issue:

1. If a mediator would be introduced, aware of w_O, it would be able to identify misunderstandings and to provide guidance for dialog efficiency.

2. The second option would be to introduce distinctive communicative acts for conveying either explanation or argument. The consequence is minimizing misunderstandings in dialog, because agents can better understand the cognitive maps of their partners.

For instance, if the agent X announces that s_1 is an explanation, its partner Y can disclose instantly its doubts about the conclusion of s_1. By updating its model $\mathcal{O}_{XY}$, the agent X can re-interpret s_1 as an argument, at this specific moment of the conversation. Thus, incorrect assumptions about accepted or doubted statements are eliminated as soon as they explicitly appear in the dialog. Moreover, people do use this kind of distinction in their discourses, when framing their speech with: *"I'll try to explain for you"*, *"One explanation is..."*, *"The main cause is" "My argument is..."*, etc. Thus, in the following we tried to build a solution for the second option.

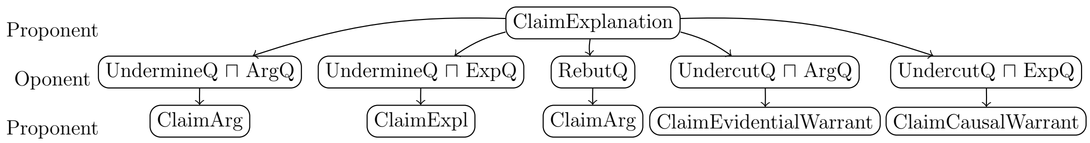


Figure 3.11: Questioning an explanation.

3.4 Dialog dynamics

We distinguish between argumentative and explicative questions.

$$ArgumentativeQ \equiv QuestionSt \sqcap \exists\, hasTopic.DoubtedSt \tag{3.7}$$
$$HowDoYouKnow \sqsubseteq ArgumentativeQ \tag{3.8}$$
$$IsItSo \sqsubseteq ArgumentativeQ \tag{3.9}$$
$$ExplicativeQ \equiv QuestionSt \sqcap \exists\, hasTopic.\neg DoubtedSt \tag{3.10}$$
$$Why \sqsubseteq ExplicativeQ \tag{3.11}$$
$$WhyDoYouConsiderThis \sqsubseteq Why \tag{3.12}$$

With an argumentative question, the agents can request evidence for a doubted conclusion. The conveyor of an argumentative question can also convey its doubts on the given topic to the receiving agents. Questions of type "How do you know" and "Is it so?" are specific cases of the argumentative ones. With an explicative question, an agent can request a cause for an accepted fact. Questions of type "Why?" are particularly considered as a request for explanation.

The argumentative agents choose among several possible moves, based on their strategies. Given a statement s in the dialog topic, the proponent may choose between direct or indirect approaches. In a direct approach the agent has the options to claim i) the statement s directly, ii) an explanation for s, iii) an argument for s. In an indirect approach, the proponent firstly claims a statement p (either evidence or cause), to test its acceptance status from the opponent's perspective. The proponent conveys the reason (argument or explanation) which links p with s only if it is accepted first by the partner.

The opponent hearing an explanation may choose between questioning part of the move which is not accepted or confirming part of on which an agreement exists. The opponent's question can be of different types (Fig. 3.11):

- $UndermineQ \sqcap ArgQ$: request evidence for the premise of explanation. Thus an argument is expected as reply.

- $UndermineQ \sqcap ExpQ$: request cause for the premise of explanation. Thus an explanation is expected as reply.

m_1	S:	Because global income of our department has increased, we have the possibility to assign more funds for teaching and research facilities.
m_2	A:	Are you sure that global income has increased?
m_3	S:	Because the number of students has increased, the partial income has increased.
m_4	A:	Partial income has been affected by the wage being increased.
m_5	S:	Is it so? My wage did not increase.
m_6	A:	The wage expenses have risen due to the recruitment of new staff in the last semester.
m_7	S:	Maybe that's why my wage did not increase.

Figure 3.12: Dialog in the education domain.

- *RebutQ*: indicate that the conclusion of explanation is not accepted. The opponent realizes that it wrongly assumed agreement on the conclusion of the explanation, when it had decided to convey that explanation. An argument is expected as reply.

- *UndercutQ* $\sqcap$ *ArgQ*: request evidence for the link between the premises and the conclusion of the explanation. An evidential warrant is expected as reply.

- *UndercutQ* $\sqcap$ *ExpQ*: request evidence for the link between the premises and the conclusion of the explanation. A causal warrant is expected as reply.

The only difference in the questioning of an argument instead of an explanation (Fig. 3.11) occurs in the rebutting of the conclusion. The *RebutQ* indicates that the proponent wrongly assumed that the conclusion was accepted/doubted by the opponent. Hence, proponent will convey an adequate reply.

Let's consider the dialog in the education domain from the Fig. 3.12, taking place between a scholar S and an administrator A. We assume that after the move m_1 both parties correctly identify the reason r_1, by interpreting the statement $\ulcorner 1 \urcorner$ as the premise and the statement $\ulcorner 2 \urcorner$ as the conclusion. The reasons appearing in the dialog and their connections are illustrated in the Fig. 3.13, while the corresponding formalization appears in the Table 3.6, where I_A and I_S are the interpretation functions for the agents A, respectively S.

No challenge being specified for $\ulcorner 2 \urcorner$ and $\ulcorner 1 \urcorner$, they are interpreted as facts by the agent S: $Fact^{I_S} = \{\ulcorner 2 \urcorner, \ulcorner 1 \urcorner\}$. Assuming that the conveyor agent S interprets the premise $\ulcorner 1 \urcorner$ as a cause (line 4 in the Table 3.6), with the causal premise $\ulcorner 1 \urcorner$ and the factual consequent $\ulcorner 2 \urcorner$, r_1 represents an explanation for the conveyor agent S (based on (3.3)), given by the $Explanation^{I_S} = \{r_1\}$.

Assuming that the agent A has a strategy to challenge all the statements that are not proved, no proof existing at this moment, the statement $\ulcorner 1 \urcorner$ is labeled as

Table 3.6: Dialog interpretation for agents A and B after move m_i

	Mv	Interpretation I_S of S	Interpretation I_A of A
1	m_1	r_1:*Reason*	r_1:*Reason*
2		$(r_1, \ulcorner 1 \urcorner)$:*hasPremise*	$(r_1, \ulcorner 1 \urcorner)$:*hasPremise*
3		$(r_1, \ulcorner 2 \urcorner)$:*hasConclusion*	$(r_1, \ulcorner 2 \urcorner)$:*hasConclusion*
4		$\ulcorner 1 \urcorner$:*Cause*	$(A, \ulcorner 1 \urcorner)$:*challenge*
5	m_2	q_1:*ArgumentativeQ*	q_1:*ArgumentativeQ*
6		$(q_1, \ulcorner 1 \urcorner)$:*hasTopic*	$(q_1, \ulcorner 1 \urcorner)$:*hasTopic*
7	m_3	(r_2, q_1):*hasStart*	
8		(r_2, r_2):*hasEnd*	
9		r_2:*Response*	
10		r_3:*Reason*	
11		$(r_3, \ulcorner 4 \urcorner)$:*hasPremise*	
12		$(r_2, \ulcorner 1 \urcorner)$:*hasConclusion*	
13		r_2:*Reason*	r_2:*Reason*
14		$(r_2, \ulcorner 3 \urcorner)$:*hasPremise*	$(r_2, \ulcorner 3 \urcorner)$:*hasPremise*
15		$(r_2, \ulcorner 4 \urcorner)$:*hasConclusion*	$(r_2, \ulcorner 4 \urcorner)$:*hasConclusion*
16		$\ulcorner 3 \urcorner$:*Cause*	
17	m_4	r_4:*Reason*	r_4:*Reason*
18		$(r_4, \ulcorner 5 \urcorner)$:*hasPremise*	$(r_4, \ulcorner 5 \urcorner)$:*hasPremise*
19		$(r_4, \ulcorner 6 \urcorner)$:*hasConclusion*	$(r_4, \ulcorner 6 \urcorner)$:*hasConclusion*
20		r_5:*ConflictRule*	r_5:*ConflictRule*
21		$\ulcorner 6 \urcorner$:*DoubtedStatement*	$\ulcorner 6 \urcorner$:*DoubtedSt*
22		$\ulcorner 4 \urcorner$:*DoubtedSt*	$\ulcorner 4 \urcorner$:*DoubtedSt*
23			$\ulcorner 5 \urcorner$:*Fact*
24	m_5	q_2:*ArgumentativeQ*	q_2 : *ArgumentativeQ*
25		$(q_2, \ulcorner 5 \urcorner)$:*hasTopic*	$(q_2, \ulcorner 5 \urcorner)$:*hasTopic*
26		$\ulcorner 7 \urcorner$:*Evidence*	$\ulcorner 7 \urcorner$:*Statement*
27		r_6:*Reason*	r_6:*Reason*
28		$(r_6, \ulcorner 7 \urcorner)$:*hasPremise*	$(r_6, \ulcorner 7 \urcorner)$:*hasPremise*
29		$(r_6, \ulcorner 5 \urcorner)$:*hasConclusion*	$(r_6, \ulcorner 5 \urcorner)$:*hasConclusion*
30	m_6	r_7:*Reason*	r_7:*Reason*
31		$(r_7, \ulcorner 8 \urcorner)$:*hasPremise*	$(r_7, \ulcorner 8 \urcorner)$:*hasPremise*
32		$(r_7, \ulcorner 5 \urcorner)$:*hasConclusion*	$(r_7, \ulcorner 5 \urcorner)$:*hasConclusion*
33			$\ulcorner 8 \urcorner$:*Evidence*
34	m_7	r_8:*Reason*	r_8:*Reason*
35		$(r_8, \ulcorner 8 \urcorner)$:*hasPremise*	$(r_8, \ulcorner 8 \urcorner)$:*hasPremise*
36		$(r_8, \ulcorner 7 \urcorner)$:*hasConclusion*	$(r_8, \ulcorner 7 \urcorner)$:*hasConclusion*
37		$\ulcorner 8 \urcorner$:*Cause*	

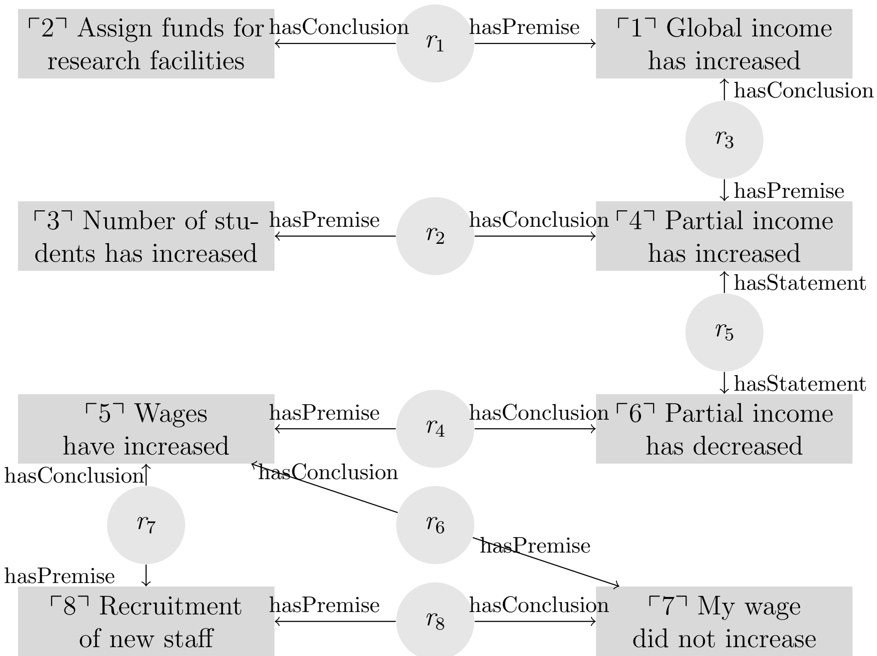


Figure 3.13: Supporting reasons in dialog.

challenged by the agent A: $challenge^{I_A} = \{(A, \ulcorner 1 \urcorner)\}$. Therefore, the statement $\ulcorner 1 \urcorner$ will be interpreted as doubted, the preconditions for uttering an argumentative question being satisfied.

The move m_2 clearly introduces some doubts regarding the statement $\ulcorner 1 \urcorner$, meaning that the agent A has no difficulty to interpret the question *"Are you sure that...?"* as an argumentative question (noted with q_1 in the line 5 of the Table 3.6). The topic of q_1 is the statement $\ulcorner 1 \urcorner$, given by: $(q_1, \ulcorner 1 \urcorner) : hasTopic$. Based on the axiom (3.46), both agents become aware that the topic $\ulcorner 1 \urcorner$ is doubted in the current dialog: $DoubtedSt^{I_S} = DoubtedSt^{I_A} = \{\ulcorner 1 \urcorner\}$, and r_1 is treated from now on as an argument by both agents.

This shift in the interpretation of r_1 from an explanation to an argument is illustrated in the first two lines of the Table 3.7. Here, while the agent S has treated r_1 as an argument from the beginning, the agent A has corrected its interpretation after the move m_2. The agent S solves the inconsistency by using the axioms $Fact \sqsubseteq \neg DoubtedSt$, $Fact^{I_S} = \{\ulcorner 1 \urcorner\}$ (after the move m_1), and $DoubtedSt^{I_S} = \{\ulcorner 1 \urcorner\}$ (after the move m_2) by removing its initially incorrect interpretation of $\ulcorner 1 \urcorner$ as a fact. After the move m_3, both agents identify the reason r_2, with premise $\ulcorner 3 \urcorner$ and consequent $\ulcorner 4 \urcorner$ (lines 13, 14, and 15 in the Table 3.6). The updated interpretation of the concept *Reason* becomes $Reason^{I_S} = Reason^{I_A} = \{r_1, r_2\}$. Given

Table 3.7: Dynamics of argument and explanation in dialog.

Move.	Ex^{I_S}	Arg^{I_S}	Ex^{I_A}	Arg^{I_A}
m_1	r_1			r_1
m_2		r_1		r_1
m_3	r_2	r_1,r_3		r_1, r_2
m_4		r_1,r_3, r_2, r_4		r_1, r_2, r_4
m_5		r_1,r_3, r_2, r_4, r_6		r_1, r_2, r_4, r_6
m_6		r_1,r_3, r_2, r_4, r_6, r_7		r_1, r_2, r_4, r_6, r_7
m_7	r_8	r_1,r_3, r_2, r_4, r_6, r_7		r_1, r_2, r_4, r_6, r_7

the interpretation of the premise $\ulcorner 3 \urcorner$ as a *Cause* by the agent S, and no challenge of the consequent, the reason r_2 is also interpreted by the agent S at this moment as an explanation: $Explanation^{I_S} = \{r_2\}$ (line 3 in the Table 3.7).

The move m_3 represents the response of agent S to the question q_1. Here $(r_2, q_1) : hasStart$ says that, triggered by the question q_1, the agent S answers with r_2, where r_2 is interpreted as a response by the agent S uttering it. Interpreted as a response by the conveyor, one of the statements in r_2 should be related to the topic questioned by q_1. Thus, according to the cognitive map of S, the consistency is assured by the hidden reason r_3. Because r_2 has doubted conclusion $\ulcorner 1 \urcorner$ and premise $\ulcorner 4 \urcorner$ represents a fact, the reason r_3 represents an argument from the viewpoint of the agent S (line 3 in the Table 3.7).

Recalling that the topic of the question q_1 is the statement $\ulcorner 1 \urcorner$ (line 6 in the Fig. 3.6), but the topic does not explicitly appear when uttering the reason r_2, it means that the hearing agent A can: i) correctly interpret r_2 as the response for q_1, but also ii) as an independent declaration in the dialog flow, with the issue posed by q_1 still open. Facing this ambiguity, one option would be to ask for clarifications regarding the membership of r_2 to the *Response* concept. The second option would be to simply react to the uttered reason r_2. The clarification may come in the form of the reason r_3, which would synchronize the cognitive maps of the two agents.

In the current dialog, A chooses to focus on one of the statements of r_2, because it is aware of the conflict r_5 regarding the statement $\ulcorner 4 \urcorner$ (line 20 in the Fig. 3.6). Defining a conflict as a reason linking two doubted statements, the statement $\ulcorner 4 \urcorner$ is categorized by the agent A as doubted, thus interpreting the reason r_2 as an argument. The newly identified argument r_2 is added to the current set of arguments of the agent A: $Argument^{I_A} = \{r_1, r_2\}$ (line 3 in the Table 3.7).

In the move m_4, the premise and the conclusion of the reason r_4 are correctly identified by both agents. The conflict between the statements "partial income has increased" and "partial income has decreased" is also clear. Both agents become aware that the consequents $\ulcorner 4 \urcorner$ and $\ulcorner 6 \urcorner$ are doubted (lines 21 and 22 in the Table 3.6). At this moment, r_4 and r_2 should also be interpreted as arguments by both parties: $Argument^{I_S} = \{r_1, r_2, r_3, r_4\}$, respectively $Argument^{I_A} = \{r_1, r_2, r_4\}$ (line 4 in the Table 4). The agent A, as the agent who proposed the argument, is not aware of any attack relation on the premise $\ulcorner 5 \urcorner$ supporting it. Therefore, according to A's current knowledge, the statement is a fact: $Fact^{I_A} = \{\ulcorner 5 \urcorner\}$ (line 23 in the Table 3.6).

The move m_5 indicates that agent S has a different opinion. Firstly, S rises the argumentative question q_2: "Is it so?". Based on q_2 and on the common knowledge in axiom (3.10), S realizes that the statement $\ulcorner 5 \urcorner$ is doubted. The agent S also provides evidence $\ulcorner 7 \urcorner$ in support of its argument r_6 (line 26 in Table 3.6).

In move m_6, knowing that the statement $\ulcorner 5 \urcorner$ is doubted, the agent A can come up with arguments supporting it. The argument r_7 is valid because its premise $\ulcorner 8 \urcorner$ is not attacked at this moment of the dialog, according to the knowledge base of the conveyor agent A. According to the current interpretation function of A, the statement $\ulcorner 8 \urcorner$ is both evidence for argument r_7 and also a fact.

In the move m_7, the agent S interprets $\ulcorner 8 \urcorner$ as a cause for why its salary did not increase (line 37 in the Table 3.6), given that the global income of the department has increased: $Explanation^{I_S} = \{r_8\}$. Depending on the next moves and possible challenge relations on $\ulcorner 8 \urcorner$ from the administrator A, the reason r_8 may shift to an argument. Note that at this moment a transfer of understanding takes place.

The following observations sum up the analysis of the dialog.

- Some reasons are explicit, and some are implicit. For instance, the implicit conflicting rule r_5 is identified by both agents, while the implicit reason r_3 is known only by the agent S (Table 3.7).

- An agent may consider that it conveys an explanation, but actually this represents an argument. (i.e. r_2 after the move m_3).

- An agent may consider that it conveys arguments, but the conveyed reason represents an explanation.

- In the light of new information, the incorrect interpretation may be updated, e.g, by uttering an argumentative question in the move m_2, the reason r_1 is interpreted by the agent S as an argument and not as an explanation, based on the initial assumptions in the move m_1, (lines 1-2 in the Table 3.7).

- Understanding can arise from conveying arguments: the explanation r_8 is constructed based on the statements from the two arguments r_7 and r_6.

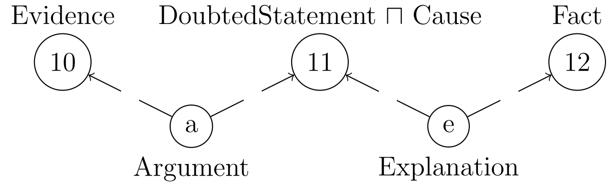


Figure 3.14: ECF pattern: conclusion of an arg. is a premise of an explanation.

3.5 Dialectical analysis of interleaved argument and explanation

In this section we model the inter-wired of arguments and explanations based on two categories of patterns: i) first order patterns, in which the conclusion of an argument or explanation is used as a premise in another one, and ii) high order patterns, in which a warrant is used to guarantee the link between the premise and the conclusion either of an argument or of an explanation.

3.5.1 First-order patterns

Arguments and explanations complete each other in natural discourses in at least three linear forms: i) the consequent of an argument is used as a cause of an explanation (the chain Evidence-Cause-Fact or ECF); ii) the consequent of an explanation is used as evidence of an argument (the chain Cause-Evidence-Doubted or CED); and iii) both circumstantial evidence and motives support the same consequent (the chain Evidence-Cause-Undecided or ECU).

ECF-pattern. A reasoning chain is an ECF pattern if the conclusion of an argument is used as a premise by an explanation. The main usage of the pattern occurs when, given a fact, a possible cause which explains that fact is doubted, so the cause needs a kind of support to decrease the level of doubt on it. Another usage of the ECF-pattern occurs when an hidden cause explains a fact, but arguments are required to support the existence of that hidden phenomena (example 10).

Example 10 (ECF pattern) *Let Abox* $= \{\ulcorner 10 \urcorner$: *Evidence,* $\ulcorner 11 \urcorner$: *DoubtedStatement,* $\ulcorner 11 \urcorner$: *Cause,* $\ulcorner 12 \urcorner$: *Fact,* a : *Reason,* e : *Reason,* $(a, \ulcorner 10 \urcorner)$: *hasPremise,* $(a, \ulcorner 11 \urcorner)$: *hasConclusion* $(e, \ulcorner 11 \urcorner)$: *hasPremise,* $(e, \ulcorner 12 \urcorner)$: *hasConclusion* $\}$. *In the reasoning pattern* $\langle \ulcorner 10 \urcorner \overset{a}{\twoheadrightarrow} \ulcorner 11 \urcorner \overset{e}{\hookrightarrow} \ulcorner 12 \urcorner \rangle$ *the conclusion of argument a is used as a premise in explanation e (see figure 3.14).*

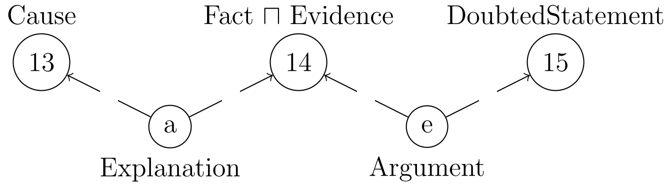


Figure 3.15: CED pattern $\langle\ulcorner 13\urcorner, \ulcorner 14\urcorner, \ulcorner 15\urcorner\rangle$: conclusion of an explanation is used as a premise in an argument.

Being the conclusion of argument a, statement $\ulcorner 11\urcorner$ is interpreted as *Doubted*. Being the premise of explanation e, the same statement $\ulcorner 11\urcorner$ is interpreted as a *Cause*. Because the same sentence acts as a conclusion of an argument and as a premise of an explanation, we need rules to model this matching.

Definition 8 *An ECF-pattern is a tuple $\langle e, dc, f\rangle$ with e interpreted as evidence, dc both as doubted statement and cause, and f as fact, constructed by the rule:*

$$\begin{aligned}\langle e, dc, f\rangle \Leftarrow\ & e{:}Evidence \wedge\ dc{:}DoubtedCause \wedge\ f{:}Fact \wedge\ a{:}Argument \wedge \\ & ex{:}eExplanation\ \wedge (a, e){:}hasPremise \wedge (a, dc){:}hasConclusion \\ & (ex, dc){:}hasPremise \wedge (ex, f){:}hasConclusion\end{aligned} \tag{3.13}$$

where DoubtedCause $\equiv$ DoubtedStatement $\sqcap$ Cause.

CED Pattern. A reasoning chain is a CED pattern if the conclusion of an explanation is used as a premise of an argument. The pattern is used when, before posting an argument, agents want to clarify the starting assumptions of the discussion (Example 11).

Example 11 (CED pattern) *Let Abox=*$\{\ulcorner 13\urcorner$*:Cause,* $\ulcorner 14\urcorner$*:Fact,* $\ulcorner 14\urcorner$*:Evidence,* $\ulcorner 12\urcorner$*:DtStatement, e:Reason,a:Reason,* $(e, \ulcorner 13\urcorner)$*:hasPrem,* $(a, \ulcorner 14\urcorner)$*:hasPre,* $(e, \ulcorner 14\urcorner)$*:hasCon,* $(a, \ulcorner 15\urcorner)$*:hasCon*$\}$

The example in figure 3.15 introduces evidence $\ulcorner 14\urcorner$ for doubted statement $\ulcorner 14\urcorner$. The explanation aims at assuring the other agent understands why statement $\ulcorner 14\urcorner$ is true. From a different perspective, the explanation aims to strength the validity of the premise of the following argument.

Definition 9 *A CED-pattern is a tuple $\langle c, fe, d\rangle$ with c interpreted as cause, fe both as fact and evidence, and d as a doubted statement, constructed by the rule:*

$$\begin{aligned}\langle c, fe, d\rangle \Leftarrow\ & c{:}Cause \wedge\ fe{:}FactiveEvidence \wedge d{:}DoubtedStatement \wedge \\ & a{:}Argument \wedge ex{:}Explanation \wedge (e, fe){:}hasCon \wedge \\ & (e, c){:}hasPre \wedge (a, fe){:}hasPre \wedge (a, d){:}hasCon\end{aligned} \tag{3.14}$$

where FactiveEvidence $\equiv$ Fact $\sqcap$ Evidence.

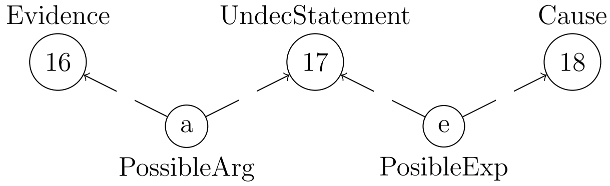


Figure 3.16: ECU pattern $\langle \ulcorner 16 \urcorner, \ulcorner 18 \urcorner, \ulcorner 17 \urcorner \rangle$: evidence and cause support the same consequent.

ECU pattern. In many situations, people use both evidence and explanations to complementary support the same consequent. Many examples come from law. Lawyers start their pledge by using the available evidence to persuade the jury about a claim which is not assumed accepted. When the jury tend to accept the claim, the lawyer provides explanations why the event took place as it really happened.

An ECU pattern occurs in two steps:

1. In the first step, evidence e is provided for supporting claim s, with s assumed undecided at the current moment,
2. In the second step, cause c is used to explain why the same statement s took place, with s assumed plausibly accepted by the audience in the light of previous evidence e (example 12).

Example 12 (ECU pattern) *Let* $Abox = \{ \ulcorner 16 \urcorner$*:CircEvidence*, $\ulcorner 18 \urcorner$*:Motive*, $\ulcorner 17 \urcorner$*:UndecStatement*, r_1*:Reason*, r_2*:Reason*, $(r_1, \ulcorner 16 \urcorner)$*:hasPre*, $(r_1, \ulcorner 17 \urcorner)$*: hasCon*, $(r_2, \ulcorner 18 \urcorner)$*:hasPre*,$(r_2, \ulcorner 17 \urcorner)$*:hasCon*$\}$

To accommodate the ECU-pattern in figure 3.16, first we need to introduce the concept of *UndecidedStatement* (in shortcut notation *UndecidedSt*), disjoint with a doubted or an accepted statement: Secondly, we refined the *ArgExp* ontology by classifying evidence (in shortcut notation *Ev*), in direct or circumstantial: A motive is a particular cause (equation 3.21).

Definition 10 *An ECU-pattern is a tuple* $\langle e, c, u \rangle$ *with e interpreted as evidence, c as a cause, and u as an undecided statement, constructed by the rule:*

$$\begin{aligned} \langle e, c, u \rangle \Leftarrow\ & e{:}Evidence \wedge c{:}Cause \wedge u{:}\ UndecidedStatement \wedge pa{:}PossibleArg \\ & \wedge pe{:}PossibleExp \wedge\ (pa, e){:}hasPremise \wedge (pa, u){:}hasConclusion \\ & (pe, c) : hasPremise \wedge (pe, u){:}hasConclusion \end{aligned} \tag{3.23}$$

$$UndecStatement \sqsubseteq Statement \quad (3.15)$$
$$UndecStatement \equiv \neg DoubtedStatement \quad (3.16)$$
$$UndecStatement \equiv \neg Fact \quad (3.17)$$
$$DirectEv \sqsubseteq Ev \sqcap \exists\, directsup.DoubtedStatement \quad (3.18)$$
$$CircumstantialEv \sqsubseteq Ev \sqcap \exists\, indirectsup.DoubtedStatement \quad (3.19)$$
$$Motive \sqsubseteq Cause \quad (3.20)$$
$$PossibleArg \sqsubseteq Reason \sqcap \forall\, hasPre.Evidence \sqcap (=1) hasCon.UndecidedSt \quad (3.21)$$
$$PossibleExp \sqsubseteq Reason \sqcap \forall\, hasPre.Cause \sqcap (=1) hasCon.UndecidedSt \quad (3.22)$$

Figure 3.17: Extended *ArgExp* ontology to acomodate the ECU pattern.

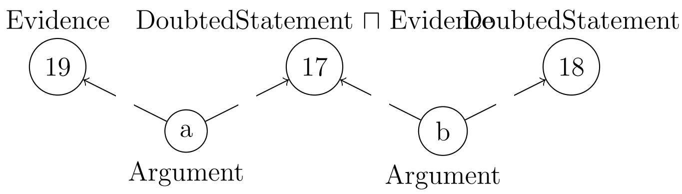


Figure 3.18: EED pattern $\langle\ulcorner 19\urcorner, \ulcorner 16\urcorner, \ulcorner 17\urcorner\rangle$: conclusion of argument a is used as a premise in argument b.

EED Pattern. A reasoning chain is an EED pattern if the conclusion of an argument is further used as a premise of an another argument. The pattern is used when an agent convey argument to support doubted evidence. First. an agent conveys an argument, considering the evidence provided is factive. If doubts are risen on the evidence, the agent can provide further evidence to strength the first piece of evidence. The concept of *doubted evidence* is meet in law, but also in scientific reasoning.

Example 13 (EED pattern) *Let* $Abox = \{ \ulcorner 19\urcorner{:}Evidence,\ \ulcorner 16\urcorner{:}DoubtedStatement,\ \ulcorner 16\urcorner{:}Evidence,\ \ulcorner 17\urcorner{:}DoubtedStatement,\ a{:}Reason,\ b{:}Reason,\ (a, \ulcorner 19\urcorner){:}hasPre,\ (b, \ulcorner 16\urcorner){:}hasPre,\ (a, \ulcorner 16\urcorner){:}hasCon,\ (b, \ulcorner 17\urcorner){:}hasCon \}$

Doubts may exist on the evidence used to suport the doubted conclusion $\ulcorner 17\urcorner$. Thus, $\ulcorner 16\urcorner$ becomes doubted evidence. The evidence $\ulcorner 19\urcorner$ supports $\ulcorner 16\urcorner$.

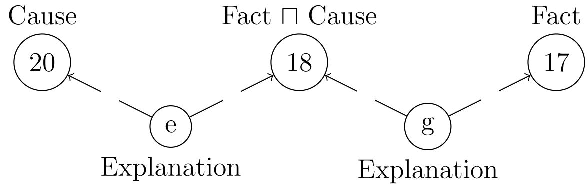


Figure 3.19: CCF pattern $\langle\ulcorner 20\urcorner, \ulcorner 18\urcorner, \ulcorner 17\urcorner\rangle$: conclusion of explanation e is used as a premise in explanation f.

Definition 11 *An EED-pattern is a tuple $\langle e, de, d\rangle$ with e interpreted as evidence, de as doubted evidence, and d as a doubted statement, constructed by the rule:*

$$\begin{aligned}\langle e, de, d\rangle \Leftarrow\ & e{:}Evidence \wedge\ de{:}DoubtedEvidence \wedge d{:}DoubtedStatement \wedge \\ & a{:}Argument \wedge b{:}Argument \wedge (a, e) : hasPremise \wedge \\ & (b, de) : hasCon,\ (a, de){:}hasPremise \wedge (b, d){:}hasCon\end{aligned} \tag{3.24}$$

where DoubtedEvidence $\equiv$ DoubtedStatement $\sqcap$ Evidence.

CCF Pattern. A reasoning chain is a CCF pattern, if the conclusion of an explanation is further used as a premise of another explanation. Thus, an agent conveys a chain of explanations such that a fact is explained by more and more refined or particular causes. The pattern is used to gradually transfer understanding for a non-primitive fact. After each explanation, a confirmation of understanding may occur (example 14).

Example 14 (CCF pattern) *Let Abox=\{ $\ulcorner 20\urcorner$:Cause, $\ulcorner 18\urcorner$:Fact, $\ulcorner 18\urcorner$:Cause, $\ulcorner 17\urcorner$:Fact, a:Reason, b:Reason, $(e, \ulcorner 20\urcorner)$:hasPremise, $(f, \ulcorner 18\urcorner)$:hasPremise, $(e, \ulcorner 18\urcorner)$:hasConclusion, $(f, \ulcorner 17\urcorner)$:hasConclusion\}*

In example 14, the conveyor of explanation receives confirmation c that the premise is accepted as *Fact* by the hearing agent. With question q_2, it requests explanations on the accepted statement $\ulcorner 18\urcorner$.

Definition 12 *A CCF-pattern is a tuple $\langle c, fc, f\rangle$ with c interpreted as cause, fc as factive cause, and f as an accepted statement, constructed by the rule:*

$$\begin{aligned}\langle c, fc, f\rangle \Leftarrow\ & c{:}Cause \wedge\ fc{:}FactiveCause \wedge f{:}Fact \wedge e : Explanation\ \wedge \\ & g : Explanation \wedge\ e, c){:}(hasPremise\ \wedge (e, fc){:}hasConclusion \\ & (g, fc){:}hasPremise \wedge (g, f){:}hasConclusion\end{aligned} \tag{3.25}$$

where FactiveCause $\equiv$ Fact $\sqcap$ Cause.

Note that, a single dialog move is necessary to construct patterns ECF and CED. Instead, for ECU, CCF and EED, three moves are needed: i) in the first one, the proponent conveys a reason; ii) in the second move, the opponent questions a part of the reason (either premise or conclusion); iii) in the third step, the proponent justifies the questioned part. Questioning a part of the reason, or requesting more information on it, may rise doubts on that part. In this case, the status of questioned part is changed from *Fact* to *Doubted*, as a postcondition of the second move.

3.5.2 High-order patterns

In natural discourses, interleaving argument and explanation often occurs in more complex patterns, such as arguing about best explanations or explaining arguments. These high-order patterns employ some form of meta-reasoning. For our task, we define meta-reasoning as a statement w in favor of a reason r, which r can be either an argumentative pair (evidence, doubted statement) or an explicative one (cause, fact). The statement w acts as a warrant strengthening the link between the premises and the conclusion of the two types reasons: argumentative and explanatory. A high order pattern *HOP* is defined as a meta-reason whose premise serves as a guarantee for a reason encapsulated with a reasoning statement:

$$HOP \equiv Reason \sqcap \forall\, hasPremise.Warrant \sqcap (=1)\ hasConclusion.ReasoningSt \quad (3.26)$$

with *ReasoningStatement* $\sqsubseteq$ *Statement*. Depending on the type of the warrant (*Cause* or *Evidence*) and on the status of the reasoning statement (*DoubtedStatement* or *Fact*), four high order patterns exist: i) meta-argument (*MA*), ii) causal argument (*CA*), iii) evidential explanation (*EE*), and iv) meta-explanation (*ME*).

Definition 13 *A causal argument CA is an argument with a causal statement as conclusion.*

$$CA \equiv Argument \sqcap (=1) hasConclusion.CausalStatement \quad (3.27)$$

with CausalStatement $\sqsubseteq$ *ReasoningStatement.*

The pattern is used when you want to argue on an explanation among others. For instance, given a set of possible explanations, you provide evidence in favor of a best explanation. With several possible explanations available, you are aware that your supported explanation may not be accepted by your partner. This means that you interpret the entire explanation (the pair cause-fact) as a *DoubtedStatement*.

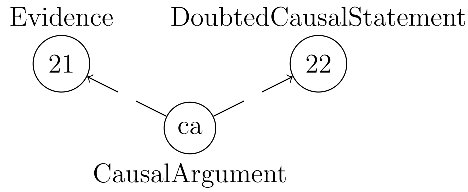


Figure 3.20: Causal argument ca: statistical evidence $\ulcorner 21 \urcorner$ warrants the doubted causal statement $\ulcorner 22 \urcorner$.

Our model reflects this because the unique conclusion of a causal argument is both a causal statement (axiom 3.27) and a doubted statement (axiom 3.2)

$$c : \mathit{DoubtedCausalStatement} \Leftarrow ca : CA \wedge (ca, c) : \mathit{hasConclusion} \tag{3.28}$$

Similarly, CA being an argument, its premise represents evidence (axiom 3.2) and being a HOP, the premise represents a warrant (definition 3.26). Thus, the premise of CA is of type evidential warrant, where $\mathit{EvidentialWarrant} \equiv \mathit{Warrant} \sqcap \mathit{Evidence}$. The following rule formalises this:

$$p : \mathit{EvidentialWarrant} \Leftarrow ca : CA \wedge (ca, p) : \mathit{hasPremise} \tag{3.29}$$

Example 15 (Causal Argument) *Let Abox=*$\{\ulcorner 21 \urcorner : \mathit{StatisticalEvidence}, \ulcorner 22 \urcorner : \mathit{DoubtedStatement}, \ulcorner 21 \urcorner : \mathit{Warrant}, \ulcorner 22 \urcorner : \mathit{CausalStatement}, (ca, \ulcorner 21 \urcorner) : \mathit{hasPremise}, (ca, \ulcorner 22 \urcorner) : \mathit{hasConclusion}\}$
labelex:causalargument

Statement $\ulcorner 21 \urcorner$ is a type of evidence $\mathit{StatisticalEvidence} \sqsubseteq \mathit{Evidence}$, while the consequent $\ulcorner 22 \urcorner$ is an explanation which is not accepted as a valid one (figure 3.20). For instance, the evidence *"We played only 5 minutes in 10 players"* may challenge the claim of $\ulcorner 22 \urcorner$. Based on definition 3.4, $\ulcorner 22 \urcorner$ is a doubted statement. Based on axiom 3.26, $CA \sqsubseteq HOP$. With further annotations, the causal statement $\ulcorner 22 \urcorner$ is interpreted as an explanation, in which the fact *"The team loosed the game"* is explained by the cause *"One of its players got the red card"*.

Definition 14 *A meta-argument MA is an argument with an argumentative statement as conclusion and warrants as premises.*

$$MA \equiv \mathit{Argument} \sqcap (=1)\mathit{hasConclusion}.\mathit{ArgumentativeStatement} \tag{3.30}$$

with $\mathit{ArgumentativeStatement} \sqsubseteq \mathit{Statement}$.

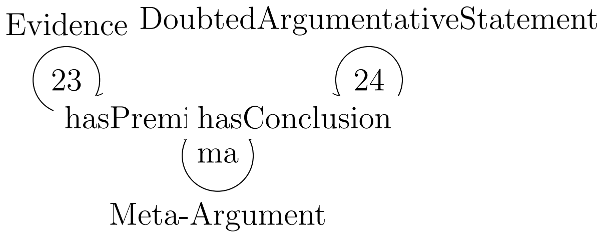


Figure 3.21: Meta-argument ma: warrant $\ulcorner 23 \urcorner$ supports the link between the evidence and claim of the argumentative statement $\ulcorner 24 \urcorner$.

Based on equations 3.2 and 3.30, the unique conclusion of a meta-argument is both an argumentative and a doubted statement.

$$c : DoubtedArgumentativeStatement \Leftarrow ma : MA \wedge (ma, c) : hasConclusion \quad (3.31)$$

In an *MA*, a warrant is interpreted as the piece of evidence which supports logical connection between a claim and its support. Note the connection between meta-arguments and the role of warrants in the Toulmin model of argumentation [162]. Similarly to a *CA*, the premise of a *MA* is an evidential warrant.

The pattern is used when the argument itself (the pair evidence-doubted statement) is not accepted by the parties, because the link between evidence and conclusion is not very clear. The warrant acts as evidence supporting the link between the premise and the conclusion of the argument. When conveying the warrant, the proponent should be aware that the supported argument is doubted.

Example 16 (Meta-argument) *Let Abox=*$\{\ulcorner 23 \urcorner : EvidentialWarrant,\ \ulcorner 24 \urcorner : DoubtedStatement,\ ma : Reason,\ \ulcorner 24 \urcorner : ArgumentativeStatement$
$(ma, \ulcorner 23 \urcorner) : hasPremise,\ (ma, \ulcorner 23 \urcorner) : hasConclusion\}$

In figure 3.21, the argumentative statement $\ulcorner 24 \urcorner$ can be further refined as an argument with evidential premise *"Global income of the university has increased"* and doubted conclusion *"We should assign more funds for research facilities"*.

Under this pattern a cause explains why there is a link between evidence and doubted conclusion. Thus, the evidential explanation pattern is used when someone wants to explain an argument.

Definition 15 *An evidential-explanation EE is an explanation with an argumentative statement as conclusion.*

$$EE \equiv Explanation \sqcap (= 1) hasConclusion.ArgumentativeStatement \quad (3.32)$$

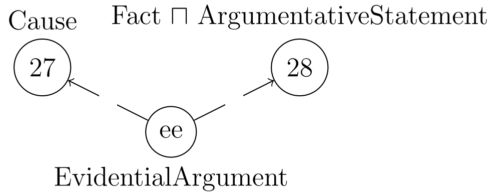


Figure 3.22: Evidential explanation *ee*: cause ⌜27⌝ explains why the argumentative statement ⌜28⌝ was accepted.

Besides being an argumentative statement (axiom 3.32), the conclusion of an evidential explanation is also a *Fact*, based on axiom 3.3, given by the rule:

$$c : FactiveArgumentativeSt \Leftarrow e : EE \wedge (e, c) : hasConclusion \tag{3.33}$$

where

$$FactiveArgumentativeSt \equiv Fact \sqcap DoubtedStatement \tag{3.34}$$

The premise of MA is a causal warrant:

$$c : CausalWarrant \Leftarrow e : EE \wedge (e, c) : hasPremise \tag{3.35}$$

with $CausalWarrant \equiv Cause \sqcap Warrant$.

One usage of the EE pattern is when, taking a decision after an argumentative process, the decisioner has to explain the decision to an audience (example 17).

Example 17 (Evidential-explanation) *Let Abox=*$\{\ulcorner 27\urcorner : CausalWarrant, \ulcorner 28\urcorner : Fact, ee : Reason, \ulcorner 28\urcorner : ArgumentativeStatement, (ee, \ulcorner 27\urcorner) : hasPremise, (ee, \ulcorner 28\urcorner) : hasConclusion\}$

The argumentative statement ⌜28⌝ (figure 3.22) can be refined as an argument with evidence *"scarcity of raw materials"* and doubted claim *"we should close the production line"*.

Meta-explanation. Under this pattern, a cause is provided in favour of an accepted explanation.

Definition 16 *A meta-explanation ME is an explanation with a causal statement as conclusion.*

$$ME \equiv Explanation \sqcap (= 1) hasConclusion.CausalStatement \tag{3.36}$$

Based on definition 3.3 and equation 3.36, the conclusion of a meta-explanation is both a *Fact* and a *CausalStatement*:

$c : FactiveCausalStatement \Leftarrow m : ME \wedge (m, c) : hasConclusion$

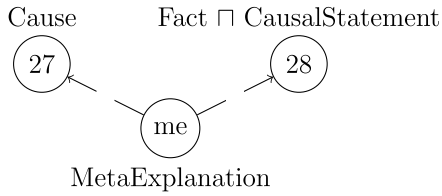


Figure 3.23: Meta-explanation *me*: cause $\ulcorner 25 \urcorner$ explains why the causal statement $\ulcorner 26 \urcorner$ is accepted .

Table 3.8: Building elements of high order patterns.

HOP	Premise	Conclusion
Causal argument	EvidentialWarrant	DoubtedCausalStatement
Meta argument	EvidentialWarrant	DoubtedArgumentativeStatement
Evidential argument	CausalWarrant	FactiveArgumentativeStatement
Meta explanation	CausalWarrant	FactiveCausalStatement

Example 18 (Meta-explanation) *Let Abox=*$\{\ulcorner 25 \urcorner : \textit{CausalWarrant}, \ulcorner 26 \urcorner : \textit{Fact}\ me : \textit{Reason}, \ulcorner 26 \urcorner : \textit{CausalStatement}, (me, \ulcorner 25 \urcorner) : \textit{hasPremise}, (me, \ulcorner 26 \urcorner) : \textit{hasConclusion}\}$

In example 18, statement $\ulcorner 25 \urcorner$ warrant explanation $\ulcorner 26 \urcorner$, (see Fig. 3.23.

The elements of the four high order patterns appear in Table 3.8. A common feature when defining high order patterns regards annotation at different levels. The statements can remain at the level of *CausalStatement* or *ArgumentativeStatement* or can be further refined as explanations, respectively arguments, by annotating their premises and conclusion accordingly.

3.6 Speech acts in description logic

The following speech acts are analyzed only from the perspective of distinguishing between argument and explanation. After modeling the communicative acts in DL, their preconditions and postconditions are formally specified.

A speech act should: i) have at least one conveyor or type agent, ii) it may have several or none hearers, iii) it has exactly one content of any type, and iv) a reply of type speech act, formalised by:

$$\begin{aligned} \textit{SpeechAct} \equiv\ & \exists\, \textit{hasConveyor.Agent} \sqcap \forall\, \textit{hasHearer.Agent} \sqcap \\ & (= \textit{1}) \textit{hasContent}.\top \sqcap \forall\, \textit{hasReply.SpeechAct} \end{aligned} \tag{3.37}$$

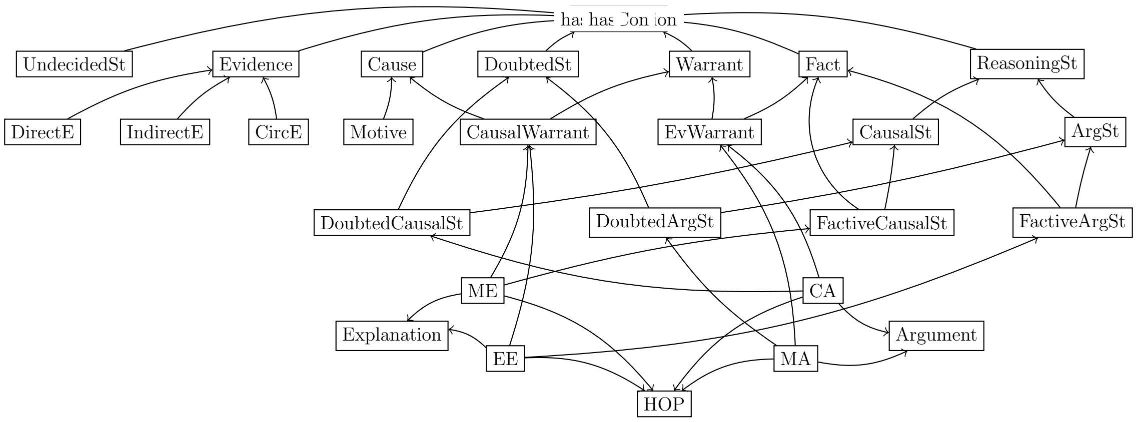


Figure 3.24: High order patterns in the *ArgExp* ontology.

To trace the connections between utterances, we introduced the *Response* concept, which is triggered by a specific speech act: $Response \equiv \exists\, hasStart.SpeechAct$ with the inverse role $hasStart^{-1} = hasEnd$. We need the following types of speech acts:

$$Claim \sqcup Question \sqcup Accept \sqcup Retract \sqsubseteq SpeechAct \tag{3.38}$$

Claiming. An agent can claim both statements and reasons. Claiming a statement means that the content of a speech act is a *Statement*, with the following replies:

$$\begin{aligned} ClaimSt \equiv \quad & Claim \sqcap (=1) hasContent.Statement \sqcap \\ & \forall\, hasReply.(AcceptSt \sqcup ExplicativeQ \sqcup ArgumentativeQ) \end{aligned} \tag{3.39}$$

More specific claims can be defined based on the type of the statement:

$$ClaimWarrant \sqsubseteq ClaimSt \sqcap (=1) hasContent.Warrant \tag{3.40}$$

Claiming a reason implies that the content of such a speech act is a reason:

$$ClaimReason \equiv Claim \sqcap (=1) hasContent.Reason \tag{3.41}$$

The conveyed reason can be argumentative or explicative:

$$\begin{aligned} ClaimArg \equiv\ & ClaimReason \sqcap (=1) hasContent.Arg \sqcap \\ & \forall\, hasReply.(Agree \sqcup UndermineQ \sqcup RebutQ \sqcup UndercutQ) \end{aligned} \tag{3.42}$$

$$\begin{aligned} ClaimExp \equiv \quad & ClaimReason \sqcap (=1) hasContent.Exp \sqcap \forall\, hasReply. \\ & (Understand \sqcup UndermineQ \sqcup RebutQ \sqcup UndercutQ) \end{aligned} \tag{3.43}$$

Questioning. Both conveyed reasons and statements can be challenged or justifications can be requested for them. Questions for statements (*QuestionSt*) focus on a specific topic, with $hasTopic \sqsubseteq hasContent$:

$$QuestionSt \equiv SpeechAct \sqcap (=1)hasTopic.Statement \tag{3.44}$$

Firstly, we distinguish between argumentative and explicative questions:

$$ArgumentativeQ \equiv QuestionSt \sqcap \exists\, hasTopic.DoubtedSt \tag{3.45}$$
$$HowDoYouKnow \sqsubseteq ArgumentativeQ \tag{3.46}$$
$$IsItSo \sqsubseteq ArgumentativeQ \tag{3.47}$$
$$ExplicativeQ \equiv QuestionSt \sqcap \exists\, hasTopic.\neg DoubtedSt \tag{3.48}$$
$$Why \sqsubseteq ExplicativeQ \tag{3.49}$$
$$WhyDoYouConsiderThis \sqsubseteq Why \tag{3.50}$$

With an argumentative question, agents request evidence for a doubted conclusion. The conveyor of an argumentative question also conveys his doubts on the given topic to the receiving agents. Questions of type "How do you know" and "Is it so?" are a particular case of argumentative ones. With an explicative question, agents request cause for an accepted fact. Questions of type "Why?" are particularly considered as a request for explanation.

A second classification regards the role of the topic statement in the argumentation chain. *Undermine* occurs when the challenged statement represents a premise. *Rebut* takes place when the challenged statement represents a conclusion [144].

$$UndermineQ \equiv QuestionSt \sqcap (=1)hastopic.Conclusion \tag{3.51}$$
$$RebutQ \equiv QuestionSt \sqcap (=1)hastopic.Premise \tag{3.52}$$

The concepts *Conclusion* and *Premise* are defined with rules:

$$c : Conclusion \Leftarrow r : Reason \wedge (r, c) : hasConclusion \tag{3.53}$$
$$p : Premise \Leftarrow r : Reason \wedge (r, p) : hasPremise \tag{3.54}$$

Undercut attacks the link between premise and conclusion [144]. Thus, in our case, an undercut question attacks a *Reason*

$$UndercutQ \equiv Question \sqcap (=1)hastopic.Reason \tag{3.55}$$

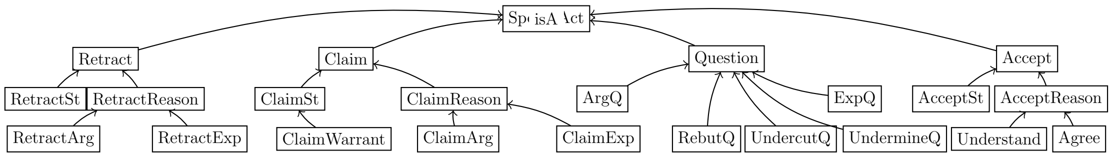


Figure 3.25: Taxonomy of speech acts for argument and explanation.

Accepting. Accepting knowledge differs from accepting understanding. Consequently, for accepting a reason there are two flavors: *agree* speech act for arguments and *understand*-like acts for explanations.

$$Agree \equiv Accept \sqcap (= 1) hasContent.Argument \tag{3.56}$$

$$Understand \equiv Accept \sqcap (= 1) hasContent.Explanation \tag{3.57}$$

where $Accept \sqsubset SpeechAct$.

The type of acceptance can be use a hint for realising how the agent has interpreting the reason: as an argument or as an explanation. When using the top level *ClaimReason* speech act (equation 3.41), by receiving an *Agree* response, the conveyor agent realises that the consequent of the reason is interpreted as *Doubted* by the respondent. By receiving an *Understand* speech act, the conveyor figures out that the consequent is accepted by the partner.

Because, unlike knowledge, understanding admits degrees [88], the *Understand* act can be further refined in complete and partial understanding: $CrystalClear \sqsubseteq Understand$, $Aha \sqsubseteq Understand$.

Retracting. Statements and reasons - either argumentative or explanatory - can be retracted:

$$RetractSt \sqcup RetractReason \sqsubseteq Retract \tag{3.58}$$

$$RetractArgument \sqcup RetractExplanation \sqsubseteq RetractReason \tag{3.59}$$

The resulted taxonomy of speech acts in DL appears in figure 3.25. The users have the possibility to convey either general or specific speech acts. The more specific the act, the easier for the partner to figure out the world of the speaker. The following semantics is assumed common knowledge for the agents.

The two cathegories of questioning presented (argumentative/explicative and rebut/undermine/undercut) are not disjoint. For instance, an argumentative question can also be an undermining one. In this case, the agent requests evidence (from the semantics of *ArgumentativeQ*) to justify the premise of the conveyed reason (from the semantics of *UndermineQ*) (line 2, column 2 in table 3.9).

Table 3.9: Questioning premises and conclusions in argument and explanation.

	Argumentative Questions	Explicative Questions
Undermine	request evidence for the premise of the reason	request cause for the premise of the reason
Rebut	request evidence for the conclusion of an explanation	request cause for the conclusion of an argument
Undercut	request evidential warrant for a reason	request causal warrant for a reason

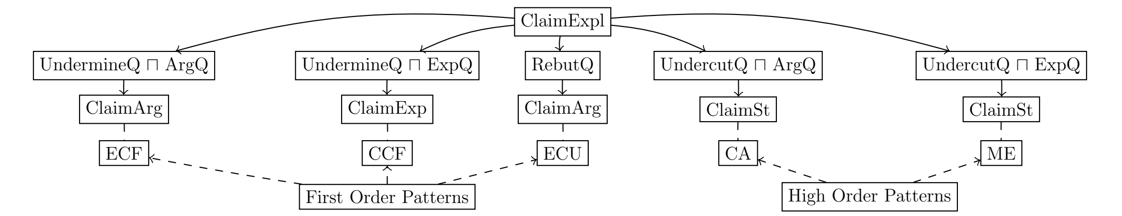


Figure 3.26: Emerging patterns from dialog games. First move: *claim explanation.*

Let agent utters an explanation and receives a rebut question in reply. It realises that the conclusion of its explanation is not accepted as fact by the other party. Which further means that the rebutting question is also an argumentative one. Consequently, the agent has to provide evidence for the conclusion, thus constructing an argument (line 3, column 2 in Table 3.9). Note that, by providing both evidence and cause for the same statement, the ECU pattern is enacted.

If an agent claims an argument and receives a rebut question, it should realise that it has wrongly labelled the conclusion as doubted. By rebutting the wrong label, its partner signals that the conclusion is factive in his world. Being also an explicative question, the partner requests a cause for the conclusion, which is now correctly labelled as accepted (line 3, column 3 in Table 3.9).

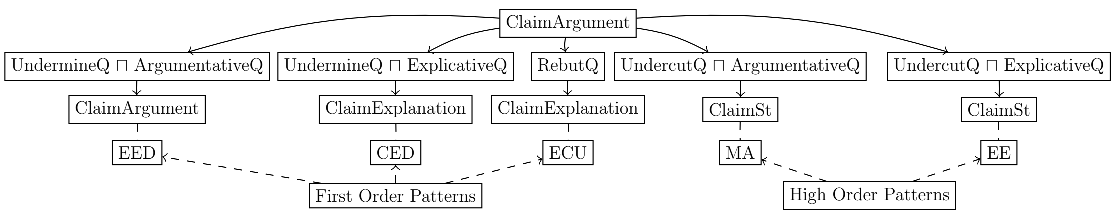


Figure 3.27: Emerging patterns from dialog games. First move: *claim argument.*

Table 3.10: Communication protocol.

Speech Acts	Possible Replies	Informal meaning
ClaimSt	AcceptSt	accept statement
	ArgumentativeQ	request for evidence
	ExplicativeQ	request for cause
	RebutQ	reject claim
ClaimArg	Agree	accept argument
	UndermineQ	challenge evidence of the argument
	UndercutQ	request warrant
	RebutQ	challenge the conclusion of argument
ClaimExp	Understand	accept explanation
	UndermineQ	request warrant
	ExplicativeQ	challenge cause
	RebutQ	challenge the conclusion as *Fact*
ArgumentativeQ	ClaimArgument	provide evidence to a *DoubtedSt*
	RetractSt	retract statement
ExplicativeQ	ClaimExplanation	provide cause to a *Fact*
	RetractSt	retract statement
UndermineQ	ClaimReason	provide support for the attacked premise
	RetractSt	retract premise
RebutQ	ClaimReason	provide support for the attacked conclusion
	RetractSt	retract conclusion
UndercutQ	ClaimSt	provide warrant for the attacked reason
	RetractReason	retract reason

3.7 Conclusions

Given the ubiquity of arguments and explanations in natural dialog, our contributions are: (i) providing guidelines to determine whether something in a dialog is an argument or an explanation [170]; (ii) modeling explanations and arguments under the same umbrella of the *ArgExp* ontology. (iii) modeling subjective perspective of agents on arguments and explanation; (iv) describing specific speech acts for conveying arguments or explanations.

By exploiting the reasoning tasks of the DL, the system we implemented is able to automatic classify arguments and explanations, based on the partial annotations from the users. The main benefit is that agents identify more quickly agreements and disagreements during dialogs.

We claim that our model may have applicability in the following areas. (i) In *legal discourses*, distinguishing between argument and explanation provides insights on the pleading games [51]. Our model allows the integration of legal ontologies for handling refined types of legal evidence. (ii) In *press articles*, our formalization is a step toward semi-automatic identification of the structure, as informally suggested in [127]. (iii) In *learning*, the use of such a system would be to structure argumentation and explanation for understanding scientific notions [35] using computer-mediated dialogs tools enriched with semantic annotation. (iv) In the *standards for dialog annotation*, by exploiting the semantics of RDF or OWL instead of XML used for the ISO 24617-2 dialog annotation standard [22], it would be easier to build applications that conform to the standard.

Our computational model may be extended in several directions. First, our approach can be seen as a starting point for defining an ontology of explanations, complementary to – and completing in our view – the AIF argumentation ontology. The second issue is how does the model fit to dialogs with more than two agents, like open discussions. What about the situation in which a mediator exists, aware of the objective world w_O? It would be interesting to compare how disagreement decreases [17] as the dialog evolves: (i) with and without a mediator and (ii) with and without explanation capabilities.

Chapter 4

Arguing in fuzzy description logic

> "A list is only as strong as its weakest link"
>
> Donald Knuth

4.1 Interleaving fuzziness and knowledge

Natural language arguments are a mixture of fuzzy linguistic variables and knowledge. This chapter focuses on modelling imprecise arguments in order to obtain a better interleaving of human and software agents argumentation, which might be proved useful for extending the number of real life argumentative-based applications. We propose Fuzzy Description Logic as the adequate technical instrumentation for filling the gap between human arguments and software agents arguments. A proof of concept scenario has been tested with the fuzzyDL reasoner.

From the practical perspective, the argumentative based applications are still very limited. One reason behind the lack of a large scale proliferation of arguments is justified by the gap between the low level expressivity and flexibility provided by the existing argumentation frameworks and the level required by the human agents. On the one hand, during the past years, the research on argumentation theory has focused on i) identifying and formalizing the most adequate technical instrumentation for modeling argumentation and ii) specifying standards for changing arguments between software agents. Defeasible logic seems to be one answer to the modelling issue [38], whilst Argument Interchange Format (AIF) ontology fulfills the requirements for arguments interchange in multi-agent systems [146]. On the other hand, argumentation schemes [93] and diagrammatic reasoning [149] based on conceptual maps have been introduced in order to provide support for human argumentation. One trend consists of developing hybrid

Table 4.1: Fuzzy arguments on junk food.

A	It may be very hard to reverse the trend of eating junk food that can be achieved by education alone.
B	It is cheap and easy for people to eat junk food, opposite to the nutrition food.
C	At the store where I shop, a candy bar costs less than a dollar and is ready to eat.
D	Candy bar can be classified as junk food.
E	Fresh fruits and vegetables tend to be inconveniently packaged and cost more.
F	Fresh fruits and vegetables can be classified as nutritious foods.
G	It is also highly profitable for manufacturers because
H	junk food has a long shelf life in the retail outlet.

approaches that combine the advantages of formal (logic-based) and informal (argumentation schemes-based, diagramming reasoning) ideas [85]. In our viewpoint the software argumentation and human argumentation should not be treated separately. Even if the software agents skills of searching, comparing and identifying fallacies in argumentation chains are quite remarkable at the propositional level, many of the argumentative domains such as legal reasoning or medical argumentation rely mostly on the interaction with the human agent, which lacks the ability to easily interpret non-linguistic arguments. Interleaving of human and software agents is useful for extending the number of real life argumentative-based applications. To meet these requirements, we used fuzzy description logic as the technical instrumentation aiming to fill the gap between software and human arguments. The DL component contributes to the current vision [146] of developing the infrastructure for World Wide Argument Web (WWAW). The fuzzy component helps agents to exploit the real arguments conveyed by humans.

4.2 Reasoning on fuzzy arguments

In human argumentation, some attacks rely on fuzzy premises. Statements like "the accused did not have a good relationship with the victim" include the fuzzy notion of *good relationship*. Also, the sentence itself may be accepted only to a certain extent, as opposed to being either accepted or not.

The example in Table 4.1 is adapted from [171]. Here A is the final conclusion. The sentence B is supported by several premises C, D, E, F, while G gives additional reasons to support B. The conclusion A contains the linguistic variable *Hard*, meaning that the point to be proved is a fuzzy concept. It also contains the modifier *very*, which can be seen as a function which alters the membership function of the fuzzy concept *Hard*. Two other fuzzy variables, *Cheap* and *Easy*, appear in the sentence B. Here, the concept *People* is linked by the role *eat* with the concept *JunkFood*. The concept *NutritionFood* can be seen as disjoint with the concept *JunkFood*. Both of them are subsumed by the general concept *Food*. But how clear is the delimitation between junk and nutrition food? The definition of junk food is applied to some food perceived to have little nutritional value, or to products with nutritional value but containing ingredients considered unhealthy:

$$JunkFood = Food \sqcap (\exists\, NutritionalValue.Little \sqcup \exists\, hasIngredients.Unhealthy) \quad (4.1)$$

Observe that in this definition there are two roles which point to the fuzzy concepts *Little* and *Unhealthy*. Let's take the common example of pizza. Can it be categorised as junk food or nutrition food? Associated with some food outlets, it is labelled as "junk", while in others it is seen as being acceptable and trendy. Rather, one can consider that it belongs to both concepts with different degree of truth, let's say 0.7 for *JunkFood* and 0.3 to *NutritionFood*.

$$Pizza \sqsubseteq JunkFood\langle 0.7\rangle, Pizza \sqsubseteq NutritionalFood\langle 0.3\rangle \quad (4.2)$$

The sentence D introduces the subconcept *CandyBar* subsumed by the concept *JunkFood*. The sentence C instantiates a particular candy bar which costs less than a dollar. The terms *Fresh* and *Inconveniently* in the sentence E are also fuzzy concepts, while the statement F introduces new knowledge: $FreshFruits \sqsubseteq NutritionalFood$, $Vegetables \sqsubseteq NutritionalFood$ The fuzzy modifier highly appears in the sentence G, and additionally, the fuzzy concept Long is introduced in the sentence H. The point that we want to bear out here is that humans consistently use both fuzzy and crisp knowledge when they convey arguments. From the technical perspective, one issue refers to what type of inference can one apply between two fuzzy arguments, e.g. B and A. What about the case in which B is supported by two independent reasons? Should one take into consideration the strongest argument, or both of concept? One advantage of fuzzy logic is that it provides technical instrumentation (Lukasiewicz semantics, Godel semantics) to handle all the above cases in an argumentative debate.

Some observations regarding the usage of the fuzzy operators in argumentation follow: The interpretation of Godel operators suits the weakest link principle in argumentation. According to this principle, an argument supported by a conjunction of antecedents of confidence α and β is as good as the weakest premise. The reason behind this principle is due to the fact that the opponent of the argument will attack the weakest premise in order to defeat the entire argument. This situation maps perfectly the semantics of the Godel operator for intersection ($\min\{\alpha, \beta\}$). Similarly, when several reasons to support a consequent are available, each having the strengths α , β , the strongest justification is chosen to be conveyed in a dialogue protocol, which can be modelled by the Godel union operator ($\max \alpha$, β).

The interpretation of Lukasiewicz operators fits better to the concept of accrual of arguments. In some cases, independent reasons supporting the same consequent provide stronger arguments in favor of that conclusion Under the Lukasiewicz semantics, the strenghts of the premises (α , β) contribute to the confidence of the conclusion, given by maxα + β,1. For instance, the testimony of two witnesses is required in judicial cases. Similarly, several reasons against a statement act as a form of collaborative defeat [8]. One issue related to applying Lukasiewicz operators to argumentation regards the difficulty to identify independent reasons. Thus, an argument presented in different forms contributes with all its avatars to the alteration of the current degree of truth. For instance, an argument subsumed by a more general argument would also contribute to the amendment of the degree of truth. Considering the argument

$$Pizza \sqcap NutritionalFood \Rightarrow AcceptableFood \tag{4.3}$$

a particular instance of pizza, belongs with a degree of $\alpha = 0.95$ to the concept of *Pizza* and with $\beta = 0.5$ to the *NutritionalFood* concept. Under the Lukasiewicz intersection operator, the degree of truth for the considered pizza to be an *AcceptableFood* is: max α + β-1,0=max0.45.0=0.45 The requirement of the accrual principle, that the premises should be independent, is violated: the degree of truth for a particular pizza to belong to the concept *AcceptableFood* is altered by the fact that the concept *Pizza* is already subsumed with a degree of 0.3 by the concept *NutritionalFood*. Thus, the description logic provides the technical instrumentation needed to identify independent justifications, whilst Lukasiewicz semantics offer a formula to compute the accrual of arguments. The accrual of dependent arguments [145] is not necessarily useless. By changing the perspective, this case can be valuable in persuasion dialogues where an agent, by repeatedly posting the same argument in different representations, will end in convincing his partner to accept that sentence. The nature of the argumentative process itself indicates that the subject of the debate cannot be easily categorised as true or false. The degree of truth for an issue and its negation are continuously changed during the lifetime of the dispute. Thus, the different levels of truthfulness (and falsity) from fuzzy logic can be exploited when modelling argumentation. Another important aspect regards the fact that argument bases are characterised by a degree of inconsistency [38]. Rules supporting both a consequent and its negation co-exist in the knowledge base. This inconsistency is naturally accommodated in fuzzy logic as the intersection between the fuzzy concept and its negation is not 0.

A_1	Normally, a small and weak person would not attack a large and strong person.
A_2	David is small and weak.
A_3	Goliat is large and strong.
C	It is implausible that David would attack Goliat.
CQ_1	Is David generally aggressive?
CQ_2	Is David a skillful fighter?
CQ_3	Is Goliat somehow clumsy?
CQ_4	Is Goliat non-aggressive?

Figure 4.1: Argument from plausible explanation.

4.3 Legal reasoning

In this legal example, one person accuses the other of assault. There had been a fight between a small and weak man on one side, and a large and strong man on the other side, and the subject is who started it. The argument of the small and weak man is whether it is plausible that he would attack the large and strong man. The plausible argument [171] is presented as an argumentation scheme in Fig 4.1. Here, we have the three premises A_1, A_2, A_3, the conclusion C, and the critical questions CQ_1-CQ_4, aiming to defeat the derivation of the consequent in case of exceptional situations. The premise A_1 contains the fuzzy qualifier normally, and thus the conclusion is subject to exceptions.

4.3.1 Computing the strength of the argument

This section shows how fuzzy description logic can be used to compute the degree of truth of the current argument. The proof of the concept scenario is formalised in the FuzzyDL reasoner (http://gaia.isti.cnr.it/*sim*straccia/) (see Fig 4.2). First, we introduce the functional roles weight and height and some constraints attached to them, such as the weight should be an integer value between 0 and 200 (lines 1 and 2). Then, we define the fuzzy concepts Small, Large *Weak*, and *Strong*, by making use of the specific fuzzy membership functions triangle and trapezoidal (lines 3, 4, 5, and 6). We continue by defining concepts such as *SmallPerson*, which is a *Person* whose height is linked to the fuzzy concept *Small* (lines 7-10).

Next, we formalize under the Lukasiewicz implication the argument that a small and weak person with an attack role towards a large and strong person leads to an implausible situation (lines 11-13). Finally, we specify instances by stating the knowledge that david is a person whose height is 161cm and his weight equals 63kg, and similarly for *goliat* (lines 14, 15). We assume that there is an attack relation from david towards goliat (line 16). When querying the reasoner, the following answers are provided under the Lukasiewicz semantics (see Table 4.2).

```
(define-concrete-feature height *integer* 0 250)
(define-concrete-feature weight *integer* 0 200)
(define-fuzzy-concept Small trapezoidal(0,250,145,150,160,165))
(define-fuzzy-concept Large trapezoidal(0,250,160,170,190,200))
(define-fuzzy-concept Weak triangular(0,200,50,60,70))
(define-fuzzy-concept Strong triangular(0,200,75,100,125))
(define-concept SmallPerson (and Person (some height Small)))
(define-concept LargePerson (and Person (some height Large)))
(define-concept WeakPerson (and Person (some weight Weak)))
(define-concept StrongPerson (and Person (some weight Strong)))
(l-implies (and SmallPerson WeakPerson
(some attack (and LargePerson StrongPerson))) ImplausibleAttack)
(instance david (and Person (= height 161) (= weight 63)) 1)
(instance goliat (and Person (= height 180)(= weight 98)) 1)
(related david goliat attack)
```

Figure 4.2: Plausible Argumentation scheme in Fuzzy Description Logic.

Table 4.2: Reasoning with the plausible argumentation scheme under the Lukasiewicz semantics.

Id	Query	fuzzyDL response
Q_1	Is david instance of SmallPerson?	0.8
Q_2	Is david instance of WeakPerson?	0.7
Q_3	Is david instance of (and SmallPerson WeakPerson))	0.5
Q_4	Is goliat instance of LargePerson?	1.0
Q_5	Is goliat instance of StrongPerson?	0.92
Q_6	Is goliat instance of (and LargePerson StrongPerson))	0.92
Q_7	Is david instance of ImplausibleAttack?	0.42

```
17 (define-truth-constant scintilaOfEvidence = 0.2)
18 (define-truth-constant resonableSuspicion = 0.4)
19 (define-truth-constant preponderenceOfEvidence=0.5)
20 (define-truth-constant clearConvincingEvidence= 0.8)
21 (define-truth-constant beyondResonableDoubt=0.95)
```

Figure 4.3: Standards of proof for accepting arguments.

```
22 (l-implies (and LargePerson StrongPerson (some attack (and SmallPerson WeakPerson))
24 (some aware LegalCase)) ImplausibleAttack)
27 (instance attackCase LegalCase)
28 (related goliat david attack)
29 (related goliat attackCase aware)
```

Figure 4.4: Shifting the burden of proof: supporting the opponent of the argument.

Based on the trapezoidal membership function of the fuzzy concept *Small* (line 3), *david* is an instance of the concept *SmallPerson* with degree α= 0.8 (query Q_1) and of the concept *WeakPerson* with β= 0.7 (query Q2). Under the Lukasiewicz semantics, *david* belongs to the intersection of the concepts *SmallPerson* and *WeakPerson* (query Q3) with the value of max$\{\alpha$ + β - 1, 0$\}$ =max$\{$0.8+0.7-1,0$\}$=0.5. Similarly, *goliat* belongs to both fuzzy concepts *LargePerson* and *StrongPerson* (query Q6) with max$\{$1.0 + 0.92 - 1, 0$\}$= 0.92. The degree of truth for david to attack *goliat* (query Q7) equals max$\{$0.5+0.92-1, 0$\}$= 0.42.

Each phase of the dispute is governed by a standard of proof, which all the conveyed arguments should meet in order to be accepted. Consider the levels of proof defined in figure 4.3. Suppose, the active standard of proof is resonableSuspicion. In this case, because david belongs to the concept ImplausibleAttack with degree of 0.42, the argument is accepted. Consequently, the burden of proof is shifted to the opponent, who has to prove that he didn't attack the other person.

4.3.2 Shifting the burden of proof

The interesting thing about this case is that the large and strong person can use a similar plausible argument to rebut the argument that made him appear guilty [171]. Thus, he claims that since it was obvious that he is a large and strong person, he would not assault the other person, especially if he was aware that the case might go to court. This argument is defined in lines 22-24 of Fig. 4.4 as a Lukasiewicz implication.

The following assertions are added to the knowledge base: Line 27 specifies that the attack event is an instance of the *LegalCase* concept. In the current phase of the dispute, the burden of proof belongs to goliat, who has to defeat the current state in which he is considered guilty of attack (line 28), while the line

```
30 (define-fuzzy-concept Long trapezoidal(0,50,5,10,20,25))
31 (define-concept Fighter (and Person (some practice FightSport)))
32 (define-concept SkilledFighter (and Fighter (some hasExperience Long)))
33 (l-implies SkilledFighter (not ImplausibleAttack))
34 (instance box FightSport)
35 (related david box practice)
36 (instance david (= hasExperience 11) 0.55)
```

Figure 4.5: Instantiating the critical question CQ2.

29 states the information that the stronger person was aware that the case could be judged in court. By asking if *goliat* is an instance of the *ImplausibleAttack* concept: the system provides based on the Lukasiewicz implication a degree of truth of 0.42. Being equal to the support of the initial argument, it means that the stronger person was able to cancel the presumption of his guilt. The expressivity of fuzzyDl allows to assign different degrees of truth both to an instance belonging to a concept, and also to roles linking instances. For example, one might say that: i) the *attackCase* will lead to a trial with a degree of truth of 0.9: *(instance attackCase LegalCase 0.9)*, or that (ii) the trust in the aware relationship between *goliat* and *atatckCase* is only 0.8: *(related goliat attackCase aware 0.8)*. For this counterargument to be successful, the lawyer must prove, beyond any reasonable doubt, that the strong person was aware that the attack ends with a trial.

4.3.3 Instantiating critical questions

The conclusion of the implausible attack is based on the current incomplete information only, meaning that no evidence addressed in the critical questions CQ1-4 has been put forward for the time being. Now, consider that the evidence related to the CQ2 has just been found out during the investigations. Specifically, it has been found that david has practised boxing for 11 years (lines 34-36 in Fig. 4.5).

Observe that the reliance on the information related to his experience is only 0.55 (line 36). From ontology, a *SkilledFighter* as a *Fighter* with long experience (line 32), where *Long* represents a fuzzy concept (line 30). The critical question CQ2 states that if the weak person is a skillful fighter, the attack on the strong person is no longer implausible (line 28). In the light of this evidence, querying the system *(min-instance? david Fighter)*, the reasoner finds that david is certainly a fighter (degree of 1.0, from lines 31, 34, and 35). He is a skillful fighter with degree of 0.55 (lines 30, 31, 36). It follows that the degree of truth for *david* to attack *goliat (min-instance? David (non ImplausibleAttack))* equals $\max\{0.55 + 1 - 1, 0\} = 0.55$, which is greater than 0.42 supporting the concept *ImplausibleAttack*.

One relevant observation is that some level of conflict is tolerated in fuzzy argumentation: an instance might belong at the same time to opposite concepts with different degrees of truth. In this line, the system can be used to identify situations in which the pieces of evidence or the ontological knowledge are inconsistent, with respect to the level of conflict accepted.

For instance, if a fact A belongs to the concept C with a degree t_1, it also belongs to the opposite concept $\neg C$ with t_2, he current system will signal that the knowledge base is inconsistent only if $t_1 + t_2 > 1$. In the current example, such a situation occurs when the level of confidence on the information related to experience (line 36) is greater then 0.58. In this case david would belong to the concepts $\neg ImplausibleAtack$ and $\neg ImplausibleAtack$ with a summed degrees of truth greater then 1.

4.4 Conclusions

Our fuzzy based approach to model argumentation is in the line of weighted argument systems of Dunne [38], aiming to provide a finer level of analysing argumentative systems. Dunne et al. have introduced the notion of inconsistency budget, which characterises how much inconsistency one is prepared to tolerate within an argumentation base. In our fuzzy approach, the tolerated inconsistency requires that the sum between the confidence in a sentence A and the confidence in its negation neg A, should be less than 1. Fuzzy knowledge bases can naturally incorporate a certain level of inconsistency, therefore no additional technical instrumentation is needed to deal with the inconsistency in argument systems.

The contributions of this chapter are: First, it proposes Fuzzy Description Logic as the adequate technical instrumentation for filling the gap between imprecise human arguments and software agents arguments. Second, we advocate the link between fuzzy reasoning (Lukasiewicz and Godel semantics) and some issues in argumentation theory (such as the weakest link principle and accrual of arguments). Also, the property of fuzzy theories to deal with inconsistency, makes them suitable to model argument bases, which are characterised by different levels of inconsistency. Finally, the paper discusses a running scenario based on plausible argumentation schemes. Additional advantages of the FDL approach are the possibility to compute the relative strength of the attack and rebuttal relationships between arguments, and the possibility to signal situations in which the fuzzy knowledge is inconsistent with respect to the level of conflict tolerated. An interesting line of future research regards the formalisation of fuzzy argumentation schemes in the Argument Interchange Format ontology [28]. Also, it would be interesting to see what advantages accrue from the argumentation based on the Description Logic restriction, rather than the full first order logic as described by Hunter and Besnard [14].

Chapter 5

Arguing in subjective logic

> “The mechanic that would perfect his work must first sharpen his tools“
>
> Confucius

5.1 Social software for climate change

Policy makers, managers and social scientists are interested in opinions of stakeholders on issues of environmental, societal and political consequences. Although social media has proven to be a precious data source for studying how people use public arena for communicating their opinions [26, 181], debate sites have not been in research focus to the same extent as other online platforms. The objective of this work is to investigate what kind of information can be extracted from individual opinions posted on debate sites. We focus on the climate change because it is a matter that interests many people who may have conflicting views and arguments. However, the method is general and can be used in other areas as well.

Debate sites are structured according to topics, (e.g. *global warming*). Anybody may post a question (e.g., *Is global warming affecting the planet?*) or a hypothesis (e.g., *Global warming is affecting the planet*), and anybody can post his or her opinion related to this question/hypothesis. In the rest of this chapter we use *hypothesis* regardless of the initiating post being in affirmative or interrogative format. The responses are votes (e.g., yes/no, pro/against or agree/disagree), which are optionally accompanied by an argumentative text. From the debate analysis perspective, the debate sites therefore possess a distinguished advantage: people's opinions about a debate topic are intrinsically labeled as pro or against, which enables automated extraction of labeled arguments.

It is not unusual that the same or similar hypotheses are discussed in more than one thread in the same debate site, even synchronously, because the debate sites do not offer a service for detecting such redundancy. For example, we noticed that *Climate change affects the earth* and *Global warming affects our planet* were debated at almost the same time within the same community. Moreover, the same hypothesis can be posted on distinct debate communities, where it may attract more (or less) negative (or positive) arguments. Thus, there is a need for computational methods to handle these cases in order to have a clearer picture on what is debated related to a topic of interest. Hence, we propose here a computational method an tool to facilitate a high level view on what is debated online.

There are several challenges in making sense of online debates. First, redundancy occurs when a person posts an existing hypothesis again, with a different wording. Hence, arguments about the same hypothesis may be spread into several threads, but debate sites do not have any mechanism to check such recurring discussions. Second, the number of responses vary significantly across topics/hypotheses which makes it difficult to compare the degree of support for two hypotheses, one with tiny and the other with massive discussions. Third, different hypotheses may be considered the same for a specific purpose, for example, of a policymaker and hence the responses to them may need to be merged. Fourth, there are several debate sites, which we call "communities", independent from each other but discussing similar or the same topics. Gathering a consolidated opinion across these communities will provide a better insight into public opinions. However, it is challenging to assess the semantic similarity between hypotheses and hence to extract collective opinions of people from distinct debate sites.

We used debate sites to extract an annotated corpus of climate change arguments in natural language. The motivation is that existing corpora for climate change are based either on media [18], or tweets [95, 133]. Both sources do introduce specific disadvantages for natural language processing. First, arguments conveyed in media are too large and sparse within an article or news. Second, arguments in tweets do not follow a proper grammar. We consider that arguments from debate sites are more adequate for natural language processing (NLP), as arguments are smaller than media documents and they are grammatically more correct than tweets. Moreover, the existing corpora contain arguments labeled as pro or against either by external human annotators or automatically (e.g. based on machine learning). Differently, the arguments in our corpus are labeled by the conveyor of the argument himself/herself. That is, the confidence in the labels is higher. Hence, such a corpus can be useful for researchers in natural language arguments or argument mining.

Our objectives are (1) to aggregate arguments posted for a certain hypothesis, (2) to consolidate opinions posted under several but related hypotheses either in the same or different debate site, and (3) to identify possible linguistic characteristics of the argumentative texts. Note that we reserve the term *aggregation* for a summary of opinions under a specific hypothesis posted in one thread, while *consolidation* is used whenever two separated threads about a topic can semantically be merged.

For the first objective, we proposed a vote-based method based on subjective logic [91]. For the second objective, we assess the semantic similarity between two hypotheses based on textual entailment [157]. For the third objective, we employ various existing lexical analysis instrumentations such as frequency analysis or readability indexes.

A social scientist using our ARGSENSE tool can obtain answers related to the following research questions:

Q_1: Are the arguers within a community apriori prone to accept or to reject a hypothesis?

Q_2: Which hypotheses are most (dis)believed or (un)popular in a community?

Q_3: Do the pro arguments have a different lexicon than the counter ones?

Q_4: Does an interrogation have more pros or more cons arguments than an affirmation?

Q_5: Are the pro arguments more readable than the con arguments?

Q_6: Is the length of hypothesis correlated with the number of arguments it receives?

Q_7: Does the formulation of the hypothesis itself (e.g., interrogative or affirmative) influence the degree of interest in the debate?

5.2 Methods and tool

We introduce here 1) the climate change argument corpus that we crawled for our experiments, and 2) the architecture of the ARGSENSE tool that we developed to facilitate the analysis of online debates.

First, we created a corpus (denoted *cc*) for the *Climate Change* domain from the three debate sites we selected: ForAndAgainst (henceforth *faa*), Debate.org (*deb*) and Debatepedia (*dbp*). All debate hypotheses discussing climate change were filtered based on the Wikipedia glossary of climate change. First, the crawled opinions are automatically structured in tuples $\langle h, t, l \rangle$, where h represents the debate hypothesis, t the argument in natural language (optional, hence may be empty), and l is the label of the vote pro (i.e., yes or agree) or cons (no or disagree)

Table 5.1: Sample of tuples $\langle h, t, l \rangle$ in the climate change corpus.

Hypothesis (h)	Argument (t)	Label (l)
Climate change is man-made.	Human carbon emissions have accelerated global warming ...	pro
Climate change is man-made.	The climate has changed through history due to natural cycles.	cons
Should government adopt emissions trading to combat global warming?	Emissions trading encourages investments in technologies.	pro

(see Table 5.1). Note that the label (pro or con) is provided by the conveyor of the argument. This label of the argument is automatically crawled from the webpage. This nice feature of the debate sites makes them an ideal source for extracting arguments already classified.

There are 1,793 hypotheses in the corpus, and total 11,653 separate responses, i.e., arguments for the whole hypotheses repertoire. The *cc* corpus was obtained by crawling three debate communities: *faa* with 142 debates containing 877 arguments, *deb* with 742 topics containing 6,026 arguments, *dbp* with 909 debates on climate change attracting 4,750 arguments. With the resulted total of 11,653 arguments, the climate change corpus is, to our knowledge, the largest corpus of labeled arguments on climate change [1].

Second, we built the ARGSENSE tool to support the analysis of people's opinions expressed in debate sites. The system is helpful for social scientists and policymakers in getting an insight into people's attitudes toward the controversial issues of worldwide interest. ARGSENSE has two architectural components relying on a *vote-based method* and *text-based methods* respectively (see Fig. 5.1).

The vote-based method takes tuples $\langle h, t, l \rangle$ by crawling the debate sites and aggregates votes l (of type pro or cons) for the same debate topic h. The aggregation is based on subjective logic. Subjective logic allows also to quantify belief and disbelief in h, but also the degree of ignorance in a community with respect to a debate topic h. The vote-based method helps a social scientist with answer to questions Q_1 and Q_2.

Text-based methods have two components: opinion consolidation and lexical analysis. By *opinion consolidation* we mean the operation of aggregating arguments of semantic similar hypotheses. The semantic similarity relation is computed using textual entailment. Textual entailment identifies hypotheses representing the same debate topic, but posted using different words (e.g. *Climate change is man-*

[1] The ARGSENSE tool and the climate change corpus are available at http://users.utcluj.ro/~agroza/projects/argclime.

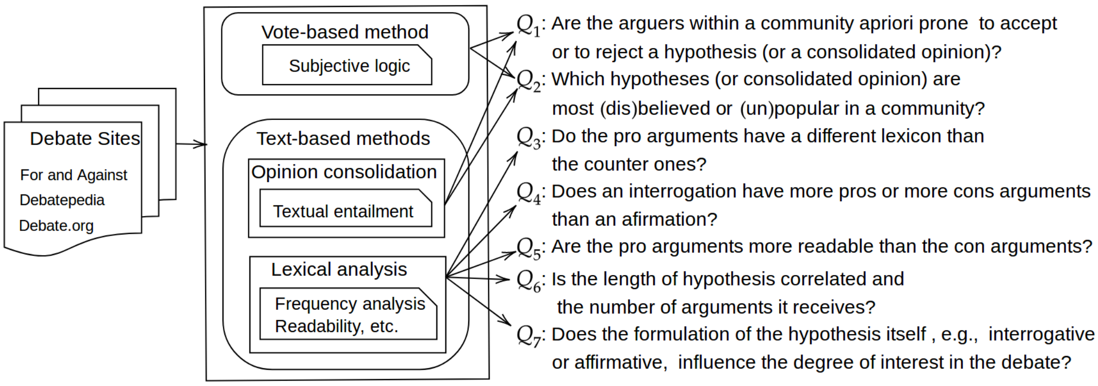


Figure 5.1: ARGSENSE investigation domain. A vote-based method based on subjective logic is proposed to rank the debate topics based on belief, disbelief and popularity in a community of arguers. An opinion consolidation method is proposed to aggregate arguments from related debate topics. This supports a more accurate view on the same questions Q_1 and Q_2. Lexical analysis uses off-shelf frequency analysis tools to support social scientists and science communicators with questions Q_3 to Q_7.

made and *Global warming is caused by humans*). Such related hypotheses can also be posted in different debate communities or posted in the same community but at different time instances. To better support the social scientist, we need to consider all the arguments posted for or against all the hypotheses representing the same debate topic. We call this process opinion consolidation. Opinion consolidation is based on textual entailment and it is the main conceptual proposal of this chapter.

Note that the vote-based method can be applied either on a single topic or on the consolidated topic that includes arguments from all related debates. In Fig. 5.1, this is illustrated by the fact that questions Q_1 and Q_2 can be applied both on a single hypothesis or on the consolidated debate topic. Throughout the paper, we use the term "aggregation" for a summary of opinions (vote-based) under a specific hypothesis, while "consolidation" is used whenever two separately posted hypotheses can semantically be merged.

Lexical analysis identifies linguistic features of argumentative texts. One can investigate if a community uses specific linguistic patterns, and whether these patterns depend on the topic or they depend on whether the discourse is supporting or countering. The methods used for lexical analysis are not new - we use readability indexes, sequential pattern mining, statistical analysis. Instead, these features help a social scientist or policymaker for answering questions Q_3 to Q_7.

5.3 Aggregation of arguments for an individual hypothesis

We describe the method for translating the individuals' arguments for a particular hypothesis in one debate site into an *aggregated opinion*.

To represent aggregated opinions we use subjective logic [91], which originally was developed for belief representation in knowledge bases. In subjective logic, an opinion ω on a given state of a system x is represented in terms of four quantities: $\omega_x = (b_x, d_x, u_x, a_x)$, where b_x represents an individual's degree of belief that the particular state x is true, d_x stands for disbelief and shows the belief that a state is false, and u_x is the uncertainty about the state. The parameter a_x is a measure of the *prior* probability of the truth value of x. In our case, the state x represents the hypothesis h for which people have provided arguments.

Differently from [91], we prefer the term *ignorance* instead of *uncertainty*, as it fits better to our task of assessing the degree in which a community is interested in a specific topic. Differently from [91], we also introduce the notion of community, to count only the arguments conveyed within a community or arguers.

The aggregated opinion of a community α about a hypothesis h is defined by:

Definition 17 *The opinion ω_h^α regarding the perceived truth value of hypothesis h by community α is a quadruple $\omega_h^\alpha = \langle b_h, d_h, i_h, a_h^\alpha \rangle$, where b_h represents the degree of belief (amount of evidence supporting h), d_h represents the disbelief (amount of evidence attacking h) and i_h represents the degree of ignorance about h with*

$$b_h + d_h + i_h = 1, \qquad \{b_h, d_h, i_h\} \in [0,1]^3 \tag{5.1}$$

The parameter a_h^α is a measure of the prior probability of the truth value of h in the community α. Hence, a_h^α is a feature of the community α. With no apriori information about α, we consider that a hypothesis has equal chances to be accepted or rejected.

In our framework, evidence for h are the arguments supporting or attacking h. For community α, let $\mathcal{A}_h^+$ be the set of arguments supporting h, and $\mathcal{A}_h^-$ the set of arguments attacking h. Let $e_h = \mid \mathcal{A}_h^+ \mid$ be the number of arguments supporting h, and $n_h = \mid \mathcal{A}_h^- \mid$ the number of arguments attacking h. The parameters b_h, d_h and i_h are computed with:

$$b_h = \frac{e_h}{e_h + n_h + 1/a_h^\alpha} \tag{5.2}$$

$$d_h = \frac{n_h}{e_h + n_h + 1/a_h^\alpha} \tag{5.3}$$

$$i_h = \frac{1/a_h^\alpha}{e_h + n_h + 1/a_h^\alpha} \tag{5.4}$$

Example 19 illustrates the opinion ω_h^α for the *h="Climate change is man-made"*.

Example 19 *Assume h="Climate change is man-made" receives* $A_h^+ = \{t_1, t_2, t_3, t_4, t_5\}$ *and* $A_h^- = \{t_6, t_7, t_8\}$*. With no apriori information about community* α *(*$a^0 = 0.5$*), we have* $b_h = 5/(5+3+2) = 5/10$*,* $d_h = 3/(5+3+2) = 3/10$*,* $u_h = 2/(5+3+2) = 2/10$*. That is the opinion* $\omega_h^\alpha = \langle 0.5, 0.33, 0.22, 0.5\rangle$*.*

With particular values for b_h, d_h or i_h, special types of opinions can be defined: i) *vacuous opinion*: $i_h = 1$ (maximum ignorance, when no argument is available for h); ii) *dogmatic opinion*: $i_h = 0$ (no ignorance; theoretically, this happens if the number of arguments is infinite); iii) *neutral opinion*: $b_h = d_h$; iv) *equidistant opinion*: $b_h = d_h = i_h$; v) *pure opinion*: $b_h = 0$ or $d_h = 0$; vi) *negative opinion*: $b_h < d_h$ (when $d_h = 1$ we have an *absolute negative opinion*); vii) *positive opinion*: $b_h > d_h$.

The fourth parameter a^α is global to the community α where h is debated. With no apriori information regarding the acceptance of h by a community of agents, a^α defaults to 0.5. More accurate representation of a^α is obtained on the basis of the distribution of positive and negative opinions. Let $\mathcal{P}^\alpha$ be the set of hypotheses in a debate community α having more positive opinions than negative ones, given by $\mathcal{P}^\alpha = \{h \in \mathcal{H}^\alpha \mid e_h > n_h\}$. Let $\mathcal{N}^\alpha$ be the set of hypotheses in the community α having more negative opinions, given by $\mathcal{N}^\alpha = \{h \in \mathcal{H}^\alpha \mid n_h > e_h\}$. With this interpretation we have:

$$a^\alpha = \frac{\mid \mathcal{P}^\alpha \mid}{\mid \mathcal{P}^\alpha \mid + \mid \mathcal{N}^\alpha \mid}, \qquad \forall\, h \in \mathcal{H}^\alpha \tag{5.5}$$

The remaining $\mathcal{E}^\alpha = \mathcal{H}^\alpha \setminus \mathcal{P}^\alpha \setminus \mathcal{N}^\alpha$ is the set of neutral hypotheses in α.

A topic h is not necessarily independent from all other topics in the same community. There can be topics claiming the contrary of h or topics claiming the same idea of h but with different linguistic expressions. Therefore, we are interested next in exploiting these inter-relations between hypotheses in α, to obtain a clearer and consolidated opinion.

5.4 Consolidation of opinions from related hypotheses

If two hypotheses are semantically close to each other, we may want to consolidate the opinions expressed for them, because it may give more information about people's attitude towards the underlying debate topic. Such hypotheses may be posted in one debate site or different ones. The question is then how to judge semantic closeness between two hypotheses. Our computational method uses three relations for semantic closeness: similarity, contradiction and entailment.

Example 20 (Similar hypotheses) *Consider g="Climate change is manmade" and h="Global warming is human made". Since g is similar to h, their supporting and attacking arguments can be aggregated.*

Let $h, g \in \mathcal{H}^{\alpha}$, $e_h = \mid \mathcal{A}_h^+ \mid$, $n_h = \mid \mathcal{A}_h^- \mid$, $e_g = \mid \mathcal{A}_g^+ \mid$, $n_g = \mid \mathcal{A}_g^- \mid$.

Definition 18 (Consolidating opinions for similar hypotheses) *If h is similar to g ($h \sim g$) then the number of positive and negative arguments for computing the consolidating opinion $\hat{\omega}_h^{\alpha}$ are:*

$$\hat{e}_h = \hat{e}_g = e_h + e_g \tag{5.6}$$
$$\hat{n}_h = \hat{e}_g = n_h + n_g \tag{5.7}$$

Example 21 (Contradictory hypotheses) *Let g="Climate change is a natural cycle". As h claims the opposite of g, the supporting arguments for h are the attacking arguments for g, while the supporting arguments for g attack h.*

Definition 19 (Consolidating opinions for contradictory hypotheses) *If h contradicts g ($h \sim \neg g$) then the number of positive and negative opinions for computing the consolidated opinion $\hat{\omega}_h^{\alpha}$ are:*

$$\hat{e}_h = \hat{n}_g = e_h + n_g \tag{5.8}$$
$$\hat{n}_h = \hat{e}_g = n_h + e_g \tag{5.9}$$

Example 22 (Entailed hypotheses)
Let k="Climate-induced changes are likely to cause effects involving many species of plants and animals" and l="Animals can be affected by climate changes". As k entails l, supporting arguments for k also support the particular claim l. But the supporting arguments for l do not necessarily support the more general hypothesis k. Instead, the attacking arguments of l also attack k. Arguments attacking k do not necessarily attack l.

Definition 20 (Consolidating opinions for entailing hypotheses) *If h entails g ($h \xrightarrow{ent} g$) then the number of positive and negative arguments for computing the consolidating opinion $\hat{\omega}_h^{\alpha}$ are:*

$$\hat{e}_h = e_h \tag{5.10}$$
$$\hat{e}_g = e_h + e_g \tag{5.11}$$
$$\hat{n}_h = n_h + n_g \tag{5.12}$$
$$\hat{n}_g = n_g \tag{5.13}$$

Three properties hold for our consolidation method:

1. less ignorance: based on the consolidated values $\hat{e_h}$ for supporting arguments and $\hat{n}_h$ for attacking arguments.

2. belief consistency: if h entails another hypothesis g, then b_h is expected to be smaller than b_g. That is: $(h \xrightarrow{ent} g) \Rightarrow (\hat{b}_h \leq \hat{b}_g)$.

3. sub-additivity of belief: if $\hat{b}_h + \hat{b}_{\neg h} < 1$.

The technical difficulty is to automatically identify these three relations: similarity, contradiction and entailment. For this task, we used the Excitement Open Platform for Textual Entailment (EOP) [139, 122]. From EOP, the Biutee algorithm [157] was preferred due to its ability to interleave knowledge from lexical resources (e.g. WordNet, VerbOcean, Wikipedia) with the language model obtained with supervised learning. Biutee converts the text into the hypothesis via a sequence of transformations. The sequence of transformations is run over the syntactic representation of the text. On the parse tree, different entailment transformations can be applied, like lexical rules (e.g. CO2 $\rightarrow$ gas) or paraphrasing rules (e.g. A affects Y $\leftrightarrow$ Y is affected by X). As these relations are usually insufficient, they are complemented with transformations from a language model. The language model is learned based on a corpus of labeled pairs of text and hypothesis. The logistic-regression is the default algorithm used by Biutee. Given all possible transformations, Biutee applies the Stern et al. search algorithm [158] to find a proof that transforms the text into the hypothesis. The availability of this proof is another reason of using Biutee in our approach.

The algorithm for consolidating opinions formalises our entailment-based method for computing consolidated opinions. The method starts by training the TE machinery with the available tuples $\langle h, t, l \rangle$ of labeled arguments. Here we exploited the advantage that the arguments are already labeled as pro or cons by their own creators. Based on the labeled pairs, we used the max entropy classification algorithm to generate a language model for climate change arguments. The resulted model contains linguistic patterns in the climate change corpus for entailment and contradiction between each hypothesis h and its supporting and attacking arguments t.

Our trick was to use this learned model to compute now the entailment relations l between pairs of hypotheses $\langle h_1, h_2, l \rangle$ instead of a pair of hypothesis and one of its arguments $\langle h, t, l \rangle$. Hence, we fed Biutee (line 11) with: i) two hypotheses h and g, ii) the model for the climate change corpus, and iii) lexical knowledge bases like WordNet or VerbOcean. Biutee will interleave domain-specific knowledge (encapsulated in the *model*) and domain-independent knowledge (i.e. WordNet, VerbOcean) to search for contradictory or entailment relations between h and g. If a contradictory relation is found, then the parameters $\hat{e}_h$ and $\hat{n}_h$ are computed based on Equations (5.8) and (5.9). If an entailment relation is found between h

and g, we check if the relation is symmetric (i.e. g entails h too). In this case (line 14), we consider the two hypotheses are semantically similar and equations (5.6) and (5.7) are applied. Otherwise, we apply equations (5.10), (5.11), (5.12) and (5.13). Note that the same hypothesis can be in various relations with other hypotheses at the same time: contradiction, entailment, or similarity.

5.5 Opinion aggregation and consolidation on the climate chance

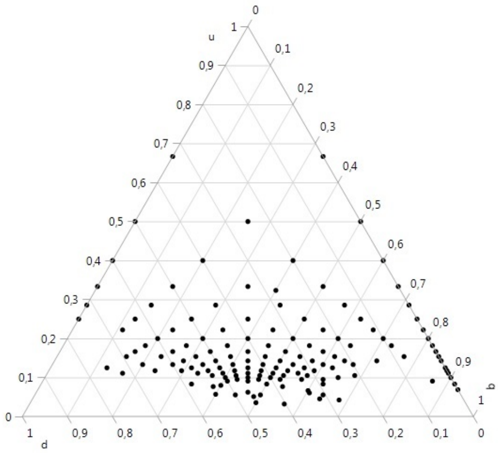
This section applies opinion aggregation and opinion consolidation on the climate change corpus. We start by ranking the topics in the corpus based on degree of belief, disbelief, or ignorance. Then we identify similar topics and we consider all their arguments in order to make a more clear picture on the ongoing debates.

Opinion aggregation

The voting based method based on subjective logic applied on the climate change corpus provides insights regarding Q_1: *Are the arguers within a community apriori prone to accept or reject a hypothesis?* In the climate change corpus a hypothesis has on average 4.5 supporting arguments and 3.62 attacking arguments. With $\mid \mathcal{P}^{cc} \mid= 943$ positive hypotheses and $\mid \mathcal{N}^{cc} \mid= 453$ we have $a^{cc} = 0.67$. On average, the degree of belief is a little larger than disbelief. Hence, members of the communities from which the arguments were collected seem to be prone to accept a given hypothesis.

All hypotheses in climate change corpus are depicted with barycentric coordinates in Fig. 5.2. Closer to the top are the hypotheses with high ignorance. Neutral opinions are on the median from the top. On the right part are positive opinions, and on the left part are the negative ones. By pressing on each of the opinion point, ARGSENSE provides details on that hypothesis or set of hypotheses.

Figure 5.2 also shows that no vacuous opinions exist in our climate change corpus. The highest degree of ignorance is 0.66, given by hypotheses with only one argument. Still, there are 194 opinions with this high degree of ignorance, representing 11% from the total of 1793 hypotheses. Among them, 88% are pure positive and 12% are pure negative. With 35 positive arguments and 25 counter arguments, the hypothesis with the smallest degree of ignorance is *"Global warming is a natural cycle"*. Note that the second hypothesis with the smallest ignorance is *"Mankind is the main cause of global warming"*, which claims the opposite of h_0. There are 57 pure negative opinions, with the highest disbelief assigned to *"Is there a climate change conspiracy behind global warming and global cooling theories?"* (5 negative arguments and 0 positive). Instead, there are 396 pure positive opinions.



No. of hypothesis	cc	$= 1{,}793$
Positive hypothesis	$\mathcal{P}^{cc}$	$= 943$
Negative hypothesis	$\mathcal{N}^{cc}$	$= 453$
Neutral hypothesis	$\mathcal{E}^{cc}$	$= 397$
Positive arguments	$\mathcal{A}^{+}_{cc}$	$= 6{,}662$
Counter arguments	$\mathcal{A}^{-}_{cc}$	$= 4{,}991$
Acceptance prone	a^{cc}	$= 0.67$
Ignorance interval	i	$\in [0.04..0.66]$
Belief interval	b	$\in [0.14..0.93]$
Disbelief interval	d	$\in [0..0.72]$

Figure 5.2: Depicting 1,793 hypotheses in the climate change corpus with barycentric coordinates. Each point depicts the set of hypotheses with the same coordinates. For instance, the opinion point $\langle 0.25, 0.25, 0.5, a^0 \rangle$ corresponds to 145 neutral hypotheses.

The pure positive opinion with the highest degree of belief is *"The Kyoto protocol would harm the American economy"*. With 22 supporting arguments and 0 against, it has a belief of 0.91. The percentage of pure opinions (25%) is quite high. No equidistant opinion exists in the corpus. There are 22% neutral opinions. Most of them are supported by one argument and attacked by one argument. There are 53% positive opinions and 25% negative opinions.

By ranking the topics based on belief, disbelief, and popularity, ARGSENSE supports Q_2 as depicted in Table 5.2. Note that the most believed hypotheses are all pure opinions, given by no counter arguments for them ($d_h = 0$). Differently, none of the four most disbelieved topics is pure, given by the existence of pro arguments for them ($b_h > 0.17$). With 35 pro and 25 con arguments, the most popular topic has an ignorance of 0.02. Note that the first two most popular hypotheses belong to the same topic - real cause of global warming - but they claim opposite statements. From the ignorance value perspective, this result is consistent with the interpretation that the cause of global warming has been the most interesting topic for the arguers. From the psychological perspective, the result might indicate that the way in which the topic of the debate is formulated influences the output of the debate: in both cases people seem to rather support the claim in the topic. The "natural cycle" hypothesis h is supported with a belief of $b_h = 0.57$, while the "human cause" hypothesis g is also supported with $b_g = 0.5$, even if g claims the opposite thing as h. One might expect that believing in h means a disbelief in g or a belief in g would be consistent with a disbelief in h.

Table 5.2: Answering to Q_2: Which hypotheses are most believed/disbelieved or popular/unpopular in a community?

Most believed hypothesis	e_h	n_h	b_h	d_h	i_h
Kyoto protocol would harm the American economy.	22	0	0.94	0	0.06
Colonizing the Moon is critical for human survival.	18	0	0.92	0	0.08
Solar shading is a just response to irreversible global warming.	18	0	0.92	0	0.08

Most disbelieved hypothesis	e_h	n_h	b_h	d_h	i_h
People can relax. Global warming is a sham.	3	13	0.17	0.74	0.09
Is cap-and-trade better at reducing emissions?	3	13	0.17	0.74	0.09
Are oil sands bad for climate change?	3	10	0.21	0.69	0.1
Is injecting sulphur dioxide into the atmosphere a good idea?	3	8	0.24	0.64	0.12

Most popular hypothesis	e_h	n_h	b_h	d_h	i_h
Global warming is a natural cycle.	35	25	0.57	0.41	0.02
Mankind is the main cause of global warming.	28	26	0.5	0.47	0.03
Should we actually have a purge?	19	18	0.49	0.47	0.04
Manmade global climate change is real and a threat.	16	18	0.45	0.51	0.04

Most unpopular hypothesis	e_h	n_h	b_h	d_h	i_h
Global warming causes earthquakes.	1	0	0.4	0	0.6
The sun causes global warming.	1	0	0.4	0	0.6
All natural disasters are related to global warming.	1	0	0.4	0	0.6

Given the nature of each debate, various factors may contribute to the above belief inconsistency at the community level.

Bottom part of Table 5.2 presents instead the hypotheses that people seem not to be interested in. There are 409 debates with only one positive argument and no attacking argument. Comparing the results of the most popular with the most ignored topics indicates that popular hypotheses are more general. Hidden variables, like time of issuing the debate, might be a cause of this lack of interest[2].

Consolidating opinions across hypotheses

The language model of climate change corpus was obtained by training the Biutee with on the *cc* corpus with 6,662 entailment pairs and 4,991 non-entailment. The entailment pairs correspond to pairs of hypotheses with supporting arguments, while non-entailment correspond to pairs of hypotheses with attacking arguments. We used the max entropy classification algorithm to generate the language model. WordNet and VerbOcean were used as external knowledge resources. From Wordnet, the following relations were considered during the search process for a useful transformation: synonym, derivationally related, hypernym, instance hypernym, member holonym, part holonym, substance meronym, entailment. Only the first sense was used for a depth limit of 2 in the Wordnet taxonomy.

Having entailment/non-etailment relations computed for the debate topics, we can now apply our consolidation method to aggregate arguments of similar topics, as exemplified in the next subsection. To illustrate the consolidation method in case of entailment consider the pair of hypotheses h=*"Mankind is the main cause of global warming"* and g=*"Global warming is real"*. h non-explicitly assumes that global warming is real and questions only its cause. Note that the assumption of h is the claim in g. Therefore we consider h entails g. In our corpus, we found that h is supported by 28 arguments and attacked by 26, while g is supported by 4 arguments and attacked by 4. That is $b_h = 0.5$ and $b_g = 0.46$. Because $h \xrightarrow{ent} g$ and $b_h > b_g$, the consistency property of belief does not hold for h and g. Instead, after applying the consolidation method in case of entailment, the consolidated belief becomes consistent and also the ignorance decreases. Based on equations (5.10) and (5.11), $\hat{e}_g = e_h + e_g = 28 + 4 = 32$ and $\hat{n}_h = n_h + n_g = 26 + 4 = 30$, while $\hat{e}_h = e_h = 28$ and $\hat{n}_g = n_g = 4$. The consolidated opinion for h is $\hat{\omega}_h^{cc} = \langle 0.47, 0.5, 0.03, 0.67\rangle$ and for g is $\hat{\omega}_g^{cc} = \langle 0.85, 0.11, 0.04, 0.67\rangle$. As the consolidated belief $\hat{b}_h < \hat{b}_g$, the belief consistency property holds between the entailing hypothesis h and g.

[2]For instance, the Marrakesh Climate Change Conference - November 2016 has not triggered many debates, as all the debates site were invaded by debates related to the USA elections.

To illustrate the sub-additive property of consolidated belief, consider the contradictory hypothesis h=*"Mankind is the main cause of global warming."* and k=*"Global warming is a natural cycle"*. Semantically, h is opposite of k. In the climate change corpus, the non-additive property does not hold for h and k ($b_h = 0.5$, $b_k = 0.57$). Instead, after applying the accrual of arguments in case of the contradictory relation, the belief becomes consistent and also the ignorance decreases. Based on equations (5.8) and (5.9), $\hat{e}_h = \hat{n}_k = e_h + n_k = 28 + 25 = 53$ and $\hat{n}_h = \hat{e}_g = n_h + e_k = 26 + 35 = 61$. the consolidated opinion for h is $\hat{\omega}_h^{cc} = \langle 0.46, 0.53, 0.01, 0.67 \rangle$ and for g is $\hat{\omega}_g^{cc} = \langle 0.53, 0.47, 0.01, 0.67 \rangle$. As for the consolidated belief $\hat{b}_h + \hat{b}_g = 0.46 + 0.53 = 0.99 < 1$, then the belief consistency property holds.

Opinion consolidation was used here as a general method for enriching the set of arguments for a given hypothesis, thus diminishing its ignorance.

5.6 Argumentative-text characteristics

Argumentative text characteristics are used by social scientists, policymakers or science communicators to better understand the communities of arguers and to design effective ways to communicate science or policies to target audience. ARGSENSE is able to analyse differences between linguistic patterns used in pro and counter arguments, to assess the correlation between the popularity of a debate with how the debate topic was posted, or to compute the readability of pro and counter arguments. We exemplify the lexical analysis of ARGSENSE by answering questions Q_3 to Q_7 on the climate change corpus.

Q_3: *Do the pro arguments have a different lexicon than the counter arguments?* Different lexicon might be an indicator to the social scientist that one party of the debate is sensible to different aspects as the other party.

To detect possible differences, we searched for the most frequent words in pro and cons arguments. For instance, if we denote by f_{20}^+ and f_{20}^- the sets of the 20 most frequent words in the set of pros and cons, we obtained $f_{20}^+ \setminus f_{20}^- = \{emissions, greenhouse, water\}$ and $f_{20}^- \setminus f_{20}^+ = \{ice, increase, opponent\}$. These results suggest that proponents of climate change are concerned with emissions and greenhouse, while the opponents rise arguments related to ice. Interestingly, the ice related counter-argument is a common misconception related to climate change [79]. ARGSENSE was able to signal that this misconception is also spread over the debate sites.

Q_4: *When does a debate get more pros than cons, when formulated as a statement or as a question?* We are interested whether posting a hypothesis in affirmative or interrogative form could modify its chances to accumulate more arguments on one side or another. In the climate change corpus, 382 affirmative hypotheses

Table 5.3: Readability indexes for pro ($\mathcal{A}^{+}_{cc}$) and against ($\mathcal{A}^{-}_{cc}$) arguments.

Readability index	Flesch Kincaid Reading Ease	Flesch Kincaid Grade Level	Gunning Fog Score	SMOG	Coleman Liau	Automated Readability
$\mathcal{A}^{+}_{cc}$	58.40	8.73	11.21	8.73	11.80	8.01
$\mathcal{A}^{-}_{cc}$	59.77	8.58	11.17	8.62	11.45	7.77

received more pro arguments and 83 of them got more counter arguments. For interrogative topics, 561 got more pros and 370 more counter arguments. Fisher's exact test indicates a very strong statistical correlation ($p < 0.0001$) between the type of hypothesis and its chances to get more positive than negative arguments. The odds ratio value for the given example is 3.04, showing that the chances to have a winner are more than three times higher when the sentence is in affirmative than in interrogative.

Q_5: *Which is the readability of the arguments conveyed in a debate?* This provides an insight on the writing and reading comprehension skills of a community of arguers. An expert in science communication uses such readability indexes to adapt its arguments to the target audience. The science communicator should balance between simplifying the text and retaining technical details.

We compare pro and counter arguments based on six readability indexes (Table 5.3). Coleman Liau and Automated Readability indexes rely on counting characters, words and sentences. The other indexes consider number of syllables and complex words. For more about readability formulas the reader is referred to [177]. No matter the readability index, the values for the positive and negative arguments are extremely similar. That is, no side uses more complex words than the other. The science communicator has to design ways to convey scientific results with the same readability indexes as the target audience or community or arguers [33].

Q_6: *Is there a correlation between the length of a hypothesis and the number of its arguments?* We investigated whether heuristics like "the shorter the hypotheses, the more arguments" can be used by a debater to decide how to formulate the debate topic. The average number of words in $\mathcal{H}^{dbp}$ is 8.67. The correlation between the length of the hypothesis and the ignorance on it is -0,01. Similarly, the average number of words in $\mathcal{H}^{deb}$ is 9.67. The correlation between the length of the hypothesis and the ignorance on it is 0.12. Based on these two low values, we can conclude that for both communities *deb* and *dbp*, the length of the hypothesis does not influence the number of arguments.

Q_7: *Does a query trigger more interest than a statement?* We are interested in analysing if a debate topic posted as *Will the planet adapt to global warming?* will attract more arguments than the version *The planet will adapt to global warming.* We evaluated the ignorance level for each hypothesis in the affirmative and

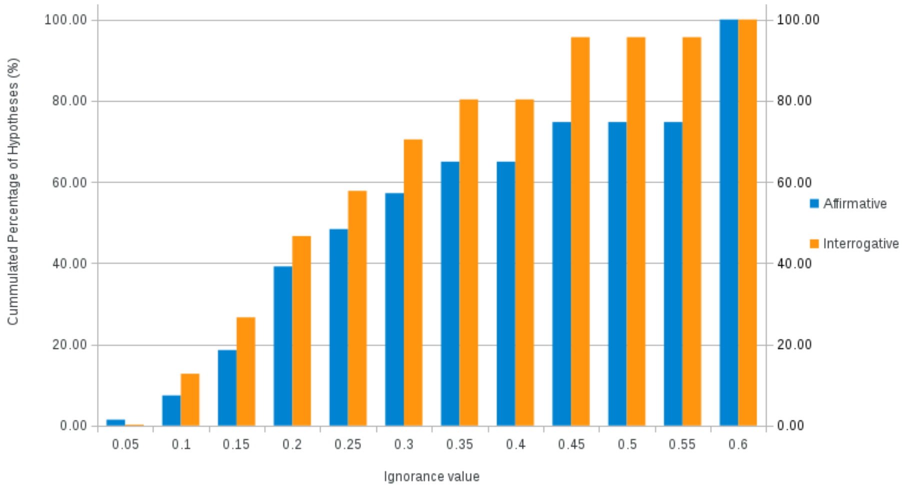


Figure 5.3: Interrogative hypotheses attract more arguments than claiming hypotheses.

interrogative form. Figure 5.3 gives the cumulated percentages of hypotheses in affirmative and interrogative format for a given ignorance threshold. For example, if we specify an ignorance threshold 0.1, we have 12.75% of interrogative hypotheses, but only 7.40% of affirmative hypotheses. The percentage of interrogative hypotheses is always higher than its affirmative counterpart, which makes us believe interrogative hypotheses have higher chances to get more intense discussion.

5.7 Conclusions

ARGSENSE analyses arguments conveyed in public arena related to climate change. The four contributions of this research are: 1) the argumentative climate change corpus, 2) the ARGSENSE social software for understanding public opinion on climate change, 3) the computational methods for aggregating and consolidating arguments, and 4) the lexical analysis of climate change arguments.

First, the climate change corpus is, to our knowledge, the largest corpus of labeled arguments on climate change.

Second, ARGSENSE aims to transform a set of arguments on climate change into usable results for policy making and climate science communication. Thus, ARGSENSE is in line with the recent trend to support scientific discovery [141] and to enhance the climate science cyber-infrastructure from ”useful to usable” decision support tools [15].

Third, social sciences need to extend their instruments to measure world-view on debate issues. Our method enhances the capabilities of social science to measure public support, disagreement, or ignorance. It employs textual entailment to find similarity, contradiction or entailment between natural arguments.

Fourth, those interested in promoting public engagement need to pay attention towards linguistic aspects of communicating climate science. Our lexical analysis performed on arguments conveyed by people found that: 1) proponents of climate change are interested in: *emissions*, *greenhouse* and *water* in positive arguments, while proponents of climate change in *ice*; 2) sequences in positive arguments do not overlap with sequences in negative arguments; 3) affirmative hypothesis have three times higher chances to win compared to interrogative hypotheses ($p < 0.0001$); 4) both pros and cons have the same readability; 5) the length of the hypothesis does not influence the number of arguments for it; 6) interrogative hypotheses have higher chances to attract more arguments than in affirmative form. Such lexical findings can be used by policy-makers and social change agents in the field of climate change to communicate more effectively to people.

Even if the confidence in computing the semantic similarity between topics is similar with the confidence of human annotators, it remains quite low: 0.65. This value is an average computed on the entire corpus. One option is to apply the opinion consolidation method only to pairs of hypotheses for which the entailment/nonentailment is computed with high confidence. Another option is to fine tune the parameters of the Biutee method related to: i) learning algorithm, ii) search process, or iii) external knowledge bases. First, there are parameters of the learning algorithm used to build the language model. We run experiments only the max entropy classification algorithm. Second, there are parameters of the search step used to build the proof for entailment or nonentailment. We used all relations from the Wordnet and a depth limit of 2 in the Wordnet taxonomy. Third, one can add domain specific knowledge bases. That is, instead of relying only on general lexical resources (Wordnet, VerbOcean, Wikipedia) one can convert domain ontologies (for climate change in our case) to a rule-based format required by TE.

Another research line to be pursued is detecting repetitive arguments, either in verbatim copies or in semantically equivalent rephrasing. Here, we considered only the number of arguments and semantic similarity between topics. To overcome this, multiple dimensions can be considered, like argument provenance or time of issue. Such direction can be integrated into the larger context of research on fake arguments, collusion of argument proponents, or on how arguments propagate in public arena or in specific communities.

Chapter 6

Arguing in justification logic

We exploit the capabilities of Justification Logic (JL) to reason about justifications. We apply JL in the context of argumentative agents. Not knowing all of the implications of their knowledge base, argumentative agents use justified arguments. The following ideas have been presented in [100]. The motivation is two fold.

6.1 Distributed justification logic

We extended the preliminary work on the application of justification logic to multi-agent systems [180, 150], by focusing on the expressiveness provided by the language in a multi-agent environment.

JL combines ideas from proof theory and epistemology. It provides an evidence-based foundation for the logic of knowledge, according to which "F is known" is replaced by "F has an adequate justification". Simply, instead of "X is known" (KX) consider $t : X$, that is, "X is known for the explicit reason t" [42]. The multi-agent version extends justified logic by introducing an index to designate agents. Consequently $t :_i F$ is read as "based on the piece of evidence t the agent i accepts F as true". The minimum justification logic is axiomatized by axioms A_0 and A_1 in Fig. 6.1. The reflection axiom A_1 is logically equivalent with $\neg F \rightarrow \neg t :_i F$, meaning that no justification t exists for a false argument.

Definition 21 *The language $\mathcal{L}$ contains proof terms $t \in \mathcal{T}$ and formulas $\varphi \in \mathcal{F}$*

$$\begin{array}{rl} t ::= & c \mid x \mid t \bullet t \mid t + t \mid !_i t \mid ?_i t \mid t \succ t \\ \varphi ::= & \gamma \mid \varphi^* \varphi \mid \neg\varphi \mid t \gg_i \varphi \mid t :_i \varphi \end{array}$$

Evidence represents a piece of knowledge which may come from communication, perception, or from a agent's own knowledge base. Following [150], we distinguish two notions of evidence: the weaker notion of admissible, relevant justification

A_0	classical propositional axioms	
A_1	$t :_i F \rightarrow F$	(weak reflexivity)
A_2	$s :_i (F \rightarrow G) \rightarrow (t :_i F \rightarrow (s \bullet t) :_i G)$	(application)
A_3	$s :_i F \rightarrow (s + t) :_i F$	(sum)
A_4	$t :_i F \rightarrow !t :_i (t :_i F)$	(proof checker)
A_5	$\neg t :_i F \rightarrow ?t :_i (\neg t :_i F)$	(negative proof checker)

Figure 6.1: Axioms of Justification Logic.

$t \gg_i \varphi$, in which the agent i admits that t is an evidence for φ, and the stronger notion of probative or factive evidence $t :_i \varphi$, in which t is strong enough making the agent i to assert φ as a fact.

Proof terms t are abstract objects that have structure. They are built up from axiom constants c, proof variables x, and agent i' operators on justifications: $\bullet$, $+$, !, ?, (see Fig. 6.1). Such an evidence-based knowledge system (EBK) is based on the following assumptions: i) all formulas have evidence ($F \rightarrow t :_i F$), ii) evidence is undeniable and implies individual knowledge of the agent (A_1); iii) evidence is checkable (A_4 and A_5); iv) evidence is monotone, new evidence does not defeat existing one (A_3) [6]. In order to adapt an EBK framework to an argumentative multi-agent system, considerations should be taken regarding the axioms A_1 and A_3, as follows.

Firstly, note that formula F is global in the multi-agent system; it is not related to any agent. In other words, if an agent $a \in \mathcal{A}$ considers t as relevant evidence to accept F, it means F should be taken as true by all the agents in $\mathcal{A}$. This not the case in real scenarios, where a different agent j might have different evidence that the opposite formula holds: $s :_j \neg F$.

Secondly, observe that the axiom A_3 encapsulates the notion of undefeasibility: if $t :_i F$, then for any other piece of evidence s, the compound evidence $t + s$ is still a justification for F. Our work regards weakening this constraint, by allowing agents to argue based on evidence with respect to the validity of a formula in a multi-agent system. This is in line with [172, 32], according to whom knowledge is incomplete and it remains open to further argument. The proposed distributed justification logic is axiomatised in figure 6.2.

E-reflexivity. A given justification of F is factive (or adequate) if it is sufficient for an agent i to conclude that F is true: $t :_i F \rightarrow F$. Knowing that the weak reflexivity property has its merits when proving theorems in justification logic, we argue it is too strong in a multi-agent environment due to:

- if the agent i has evidence t for F it does not necessarily mean that F is a fact, for other agents may provide probative reasons for the contrary;

A_0	classical propositional axioms	
A'_1	$t :_{\mathcal{E}} F \rightarrow F$	(e-reflexivity)
A'_2	$s :_i (F \rightarrow G) \rightarrow (t :_j F \rightarrow (s \bullet t) :_k G)$	(distributed application)
A'_4	$t :_i F \rightarrow !^j t :_i (t :_i F)$	(positive proof checker)
A'_5	$\neg t :_i F \rightarrow ?^j t :_i (\neg t :_i F)$	(negative proof checker)
A'_6	$s :_i F \wedge t :_j F \rightarrow (s + t) :_i F,\ s + t \succ t$	(accrual)
A'_7	$F \rightarrow t :_i F$	(internalization)

Figure 6.2: Distributed Justification Logic.

- the agents accept evidence based on different proof standards: whilst a credulous agent can have a "scintilla of evidence" standard, its partner accepts justification based on the "behind reasonable doubt" standard;
- the same evidence is interpreted differently by the agents in the system.

In our approach, a formula F is valid if all the agents in the system have justifications for F (their own or transferred from the other agents). The E-reflexivity axiom is read as: if every agent in the set E has justifications for F, F is a fact.

Distributed Application. In justified logic, the application operator takes a justification s of an implication $F \rightarrow G$ and an evidence t of its antecedent F, and produces a justification $s \bullet t$ of the consequent G [8]. In the existing multi-agents versions, the i index is introduced to represent the agent i, with the obvious meaning: if the agent i accepts the implication $F \rightarrow G$ based on s and F based on t, then agent i accepts G based on evidence $s \bullet t$ (axiom A_1). In a multi-agent setting, agents can construct their arguments based on justifications or evidence provided by their partners. Reasoning can also be performed based on the fact that the other agents rely their knowledge on a specific piece of evidence. The proposed generalised application operator A'_1 allows agent k to construct its own evidence $s \bullet t$ based on the facts that: i) the agent i has accepted the justification s as probative for $F \rightarrow G$, and ii) the agent j has accepted the evidence t to be sufficient to accept F.

Example 23 *Assuming that agent a after some symptoms visits the physician p. Based on the consultation c, the physician decides there is evidence for the disease G and requests some analysis t to investigate F, which is needed to confirm the hypothesis ($F \rightarrow G$). Agent a gets confirmation from the laboratory expert e. Consequently, it has the justification $c \bullet t$ to confirm G. The distributed application operator is instantiated as follows:*

$$c :_p (F \rightarrow G) \rightarrow t :_e F \rightarrow (c \bullet t) :_a G$$

From the functional programming perspective, assuming that $\rightarrow$ is right associative, the distributed application operator has the following meaning: when an agent p provides a justification for $F \rightarrow G$, a function is returned which waits for the evidence t confirming F in order to output the justification $c \bullet t$ for G.

Recall, that $t :_i \varphi$ represents strong evidence, opposite to weak evidence $t \gg_i \varphi$.

Example 24 *Consider that the laboratory analysis t confirming F may be contaminated, so the agent e accepts only as admissible the piece of evidence t. The corresponding expressiveness holds: "If you provide me defeasible evidence about F, I will have only admissible evidence about G:*

$$c :_p (F \rightarrow G) \rightarrow t \gg_e F \rightarrow (c \bullet t) \gg_k G$$

The subjectivity about evidence can be also expressed: what is admissible for one agent is probative for the other one. In this case the agent a considers t as strong enough for F, the evidence transfer being modelled as

$$t \gg_e F \rightarrow t :_a F$$

Example 25 *Assuming that the agent p is the same with e in A'_2, a simple justification based dialogue takes place: "I have a justification for $F \rightarrow G$. When you provide me evidence or symptom of F, I will have a justification for G".*

$$s :_i (F \rightarrow G) \rightarrow t :_j F \rightarrow (c \bullet t) :_i G$$

Positive proof checker. Justifications are assumed to be verifiable. A justification can be verified for correctness, by the other agents or by the agent who conveyed it. $t :_i F \rightarrow !^j t :_i (t :_i F)$ is read as: if t is a justification for F accepted by the agent i, the agent j can check that piece of evidence. In case the agent checks itself ($j = i$) we have *positive introspection*: $t :_i F \rightarrow !^i t :_i (t :_i F)$. It assumes that given evidence t for F, the agent i is able to produce a justification $!t_i^i$ for $t :_i F$. Thus, each justification has its own justification.

From the dialogical perspective, the positive proof checker is used to request for details why a formula is accepted based on a specific piece of evidence. The term $!^j t$ describes the agents i's evidence justifying $t :_i F$. Often, such meta-evidence has a physical form, such as a reference or email. Observe that the justification can be adapted to the agents who requested them: $!^j t :_i (t :_i F) \neq !^k t :_i (t :_i F)$. Here, the terms used by the agent i to describe the justification t for accepting F may not be equal $!^j t \neq !^k t$.

Negative proof checker. The negation in our framework is interpreted as:

$$\neg t :_i F \sim t \textit{ is not a sufficient reason for agent } i \textit{ to accept } F$$

If t is not sufficient evidence for agent i to accept F, given by $\neg t :_i F$, the agent should have a justification for this insufficiency: $\exists\, q \in \mathcal{T}_i$ such that

$$\neg t :_i F \rightarrow q :_i \neg t :_i F$$

The operation ? gets a proof t and a formula F, and outputs a proof q justifying why p is not admissible evidence for F: $? : \mathit{prof} \times \mathit{proposition} \rightarrow \mathit{proof}$. In case the agent checks itself ($j = i$) we have *negative introspection*: $\neg t :_i F \rightarrow ?^i t :_i (\neg t :_i F)$

Accrual. The axiom A_6' says that if agent i has proved s for F and agent j has evidence t for the same F, the joint evidence $s + t$ is a stronger evidence for the agent i to accept F, modelled by the preference relation $\succ$ over justifications: $t + s \succ t$. When $i = j$, the same agent has different pieces of evidence supporting the same conclusion.

Internalisation. The internalisation property assumes that formulas should be verifiable. It says that if F is valid, then there is a at least one agent i, which has accepted F based on the evidence t. From the argumentation viewpoint, every argument should have a justification in order to be supported. Consequently, self defending arguments are not allowed.

Note that, if F is a formula and t is an acceptable justification for agent i then $t :_i F$ is a formula. Thus, relative justifications of the form $s :_i (t :_j F)$ are allowed, where agent i has evidence s that agent j has evidence t for F. Similarly, the formula $t :_i F \rightarrow s(t)_i G$ says that: if t is agent i's justification for F, then $s(t)$ is agent i's evidence for G, where the argument t is inserted in the right place of argument $s(t)$. This *proof-based evidence* for G is similar to have deductive argumentation supporting G [150].

Two rules of inference hold: $F, F \rightarrow G \vdash G$ (Modus Ponens) and $\vdash c : A$ (Axiom Internalization), where A is an axiom and c is a constant. Similarly to [180] we assume that axioms are common knowledge.

6.2 Argumentation framework

Note that having evidence for something is different from convincing someone of that issue. The justified claim can be rejected if it is too discrepant with the agent knowledge base or due to the lack of understanding of the evidence.

Definition 22 *An argument is a piece of reasoning* $j :_i F$ *in which the support* j *represents a proof term intended by agent* i *to provide evidence for accepting the doubted conclusion* F.

Differently from the classical definition of an abstract argument, where the support represents a set which is minimal and without structure, here the support j represents an explicit proof term facilitating access to the reasoning chain of the agent conveying the argument.

Example 26 *Bird is the justification of agent* i *for the sentence Fly, given by* $bird :_i Fly$. *The penguins, which are birds (*$penguin \to bird$*), represent an exception, which according to agent* j*, blocks the acceptability of evidence bird as being enough for the sentence Fly. The application operator is used to model the exception:* $[penguin \cdot (penguin \to bird)] :_j \neg bird :_i Fly$.

An argument A is consistent with respect to an evidence t if A does not contradict any evidence in t. We say that a piece of evidence t does not defeat evidence s of an agent i if $s :_i F \to (s + t) :_i F$.

Definition 23 (Undercutting defeater) *The evidence* t *is an undercutting defeater for* F *justified by* s *if the joint evidence* $s + t$ *does not support* F *any more. Formally:* $s :_i F \to \neg(s + t) :_i F$

Property 1 (Justified undercutting defeater) *Note that the undercutting defeater is an implication, which is a formula in justified logic. So, based on the internalisation axiom* A'_7*, it should have a justification:* $q :_i (s :_i F \to \neg(s+t) :_i F)$. *Informally,* q *is agent's* i *justification why the piece of evidence* t *attacks evidence* s *in the context of* F *formula.*

Example 27 *Consider the dialogue in figure 6.3. Here,* m_1 *represents Adam's justification for going to the movie:* $m_1 :_A Go$. *This information (*m_1*) combined by Eve with the fact that she likes comedies (*m_2*) is strong enough for Eve to accept the invitation:* $(m_1 + m_2) :_E Go$. *However, she checks for evidence that movie is a comedy:* $!^E m_1 :_A m_1 :_A Go$. *For Eve, the new evidence* m_3 *is the undercutting defeater for the* m_1 *justification:*

$$(m_1 + m_2) :_E Go \to \neg(m_1 + m_2 + m_4) :_E Go$$

Adam requests some justification, where the complete formulation "Why, given that you like comedies, the movie is a comedy you decided to come, but when you found that John told me this you have changed your mind?" is represented as

$$!^A q :_E (m_1 + m_2) :_E Go \to \neg(m_1 + m_2 + m_4) :_E Go$$

(m_1)	Adam:	The movie is a comedy. We should go.
(m_2)	Eve:	I like comedies. We can go. How do you know that is it a comedy?
(m_3)	Adam	John told me.
(m_4)	Eve:	Then we should consider something else.
(m_5)	Adam:	Why?
(m_5)	Eve:	You know John, he laughs from everything.
(m_6)	Adam:	This usually happens. But it is not the case here.
(m_7)	Eve:	How is that?
(m_8)	Adam:	John told me the plot and it is really funny.
(m_9)	Eve:	You convinced me. Let's go then.

Figure 6.3: Justified undercutting defeater.

where $q = (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E Go$ *is the justification that should be provided by Eve to Adam for the above implication. Eve's justification comes from the* m_5 *message:*

$$m_5 :_E (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E$$

Next, Adam confirms that this usually happens

$$m_5 :_A (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4 \gg_E$$

but he does not consider the justification m_5 *as strong enough:*

$$\neg m_5 :_A (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E$$

On Eve's request for justification, Adam provides the m_8 *message:*

$$m_8 :_A \neg m_5 :_A (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E Go$$

which is eventually accepted by Eve:

$$m_8 :_E \neg m_5 :_A (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E Go$$

. According to axioms A_1' *and*

$$m_8 :_{\mathcal{E}} \neg m_5 :_A (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E Go$$

. one can state that:

$$\neg m_5 :_A (m_1 + m_2) :_E Go \rightarrow \neg(m_1 + m_2 + m_4) :_E Go$$

which means that everybody agrees the evidence m_5 *is not strong enough to defeat the Go formula supported by* m_1 *and* m_2*.*

(m_1)	Adam:	The movie is a comedy. We should go.
(m_2)	Eve:	I like comedies. We might go. When does it start?
(m_3)	Adam	At 6'o clock.
(m_4)	Eve:	We cannot then.
(m_5)	Adam:	But why?
(m_5)	Eve:	I have to be home at 9'o clock.
(m_6)	Adam:	This is not a problem.
(m_7)	Eve:	How is that?
(m_8)	Adam:	The movie takes only 2 hours.
(m_9)	Eve:	Perfect. Let's go then.

Figure 6.4: Justified rebutting defeater.

Definition 24 (Rebutting defeater) *The evidence t is a rebutting defeater for F if it is accepted as a justification for $\neg F$.*

Example 28 *Consider the dialogue in figure 6.4. Here, Eve accepts as joint evidence m_1 and m_2 for the possibility to go: $(m_1 + m_2) \gg_{Eve} Go$. The evidence m_3 is a rebuttal defeater for attending the movie: $m_3 :_E \neg Go$. When Adam asks for clarifications ($?^A m_3 :_E m_3 :_E: \neg Go$) the m_5 message is provided: $m_5 :_E m_3 :_E \neg Go$, which is not considered by Adam as strong enough $\neg m_5 :_A (m_3 :_E \neg Go)$. When asking for evidence $?^E \neg m_5 :_A \neg m_5 :_A (m_3 :_E \neg Go)$, the m_8 justification is given: $m_8 :_A (\neg m_5 :_A (m_3 :_E \neg Go))$, which is accepted by Eve too: $m_8 :_E (\neg m_5 :_A (m_3 :_E \neg Go))$.*

The following definition follows the Walton's [172] formalisation of knowledge.

Definition 25 *Knowledge represents justified acceptance of a proposition based on evidence and supported by rational argumentation to a specified standard of proof.*

This definition is accommodated in our framework by introducing an index representing the active standard of proof during the debate:

$$t :_i^\beta F \simeq i \text{ accepts } F \text{ based on the evidence } t \text{ under the standard of proof} \beta$$

An example of such standards occurs in trials: scintilla of evidence, preponderance of evidence, clear and convincing evidence, or behind reasonable doubt.

Example 29 *Consider two standards of proof scintilla of evidence (α) and preponderance of evidence (β). The piece of evidence $false\_alibi :_j^\alpha Guilty$ is accepted by the judge j as a justification for Guilty when the active standard of proof is α, but the same justification is not enough to support guiltiness under the β standard: $\neg false\_alibi :_j^\beta Guilty$.*

We assume that: justifications are abstract objects which have structure, and agents do not lose or forget justifications [8].

The omniscience problem. The agents cannot always be expected to follow extremely long or complex argumentation chains [172], even if argumentation formalisms such as hierarchical argumentation frameworks [131], or the AIF ontology [147] do not specify any constraint on the size of argument. A constraint is imposed on proof terms that are too complex with respect to the number of symbols or nesting depth. In justification logic, the complexity of a term is determined by the length of the longest branch in the tree representing this term. The size of terms is defined in a standard way: $\mid c \mid = \mid x \mid = 1$ for any constant c and any variable x, $\mid (t \bullet s) \mid = \mid (t + s) \mid = \mid t \mid + \mid s \mid + 1, \mid !t \mid = \mid t \mid + 1$.

Lemma 1 *For each justified argument conveyed by agent i to j, agent j has a justification for accepting the argument or a justification for rejecting the argument:*

$$t :_i A \rightarrow s :_j A \vee r :_j \neg A$$

Preference over justifications. Agent i prefers evidence t_1 over t_2 to justify F is represented as $t_1 \succ t_2 :_i F$. It follows that at least t_1 should be an acceptable justification for F.

$$(t_1 \succ t_2) :_i F \rightarrow t_1 :_i F$$

The piece of evidence t_2 can be connected to F in the following ways: i) t_2 is also an accepted justification of F ($t_2 :_i F$), ii) t_2 is justification for the opposite formula $\neg F$, iii) t_2 is independent of the claim F.

Agent j can check why does his partner i prefer t_1 over t_2 to justify F:

$$!(t_1 \succ t_2) :_j (t_1 \succ t_2) :_i F$$

Agent i prefers justification t_1 over t_2 in the context of F based on evidence s:

$$s :_i (t_1 \succ t_2) :_i F$$

Agent i has a justification s why his partner j prefers evidence t_1 over t_2 as justification for F:

$$s :_i (t_1 \succ t_2) :_j F$$

Preference change over evidence can not be expressed without temporariness. Based on the accrual axiom the following implications hold:

$$s :_i F \wedge t :_i F \rightarrow t + s \succ t :_i F, s :_i F \wedge t :_i F \rightarrow t + s \succ s :_i F$$

Assume that x is i's justification of A, whilst y is j's evidence regarding B.

Lemma 2 *A distributed proof term $s(x, y)$ can be constructed representing common justification accepted by the two agents to prove the intersection between A and B. Formally:*

$$x :_i A \wedge y :_j B \rightarrow s(x, y) :_{ij} (A \wedge B)$$

Communication of justifications. The following proof terms can be joint to express complex argumentative debates:

- Agent j has a justification r proving that agent i is inconsistent: $r :_j (t :_i F \wedge s :_i \neg F)$.
- Agent j has evidence showing that two agents disagree: $r :_j (t :_i F \wedge s :_k \neg F)$.
- The piece of evidence t does not defeat agent's i evidence s about F: $s :_i \rightarrow (s + t) :_i F$.
- Evidence conversion: $t :_i F \rightarrow t :_j F$. In other words, agent j trusts agent i's evidence regarding F.

6.3 Running scenario: arguing on debate sites

The proof of concept scenario is a debate regarding the issue "It is reasonable to accept the theory of evolution"[1]. Sets of arguments are exchanged during rounds between the instigator i and the contender c. Most of the burden of proof is carried by the instigator, however, the contender must defend his position that evolution is untrue ($\neg Evolution$).

Round 1. The instigator starts by stating the claiming formula, noted as *Evolution*. Based on the axiom A_7' agent i should have evidence t to support his claim, under the standard of proof "preponderance of evidence" (p). Formally,

$$Evolution \rightarrow t :_i^p Evolution$$

The contender accepts the challenge by stating his position "Evolution doesn't exist, but can you convince me?. This two pieces of information are formalised in distributed justified logic as "$\neg Evolution$, respectively

$$!^c t :_i : t :_i Evolution$$

in which the agent c requests agent i to provide justifications.

[1]Adapted from http://www.debate.org/debates/It-is-reasonable-to-accept-the-theory-of-evolution/1/

Round 2. The instigator develops his speech by stating that: "As an anthropology student, interested in human evolution, I have some education in this subject", coded as $m_1 :_i (AntStud \rightarrow Education)$ and $m_2 :_i AntStudent$. Based on the application operator, a justification is derived from the sentence *Education*:

$$m_1 :_i (AntStud \rightarrow Education) \rightarrow m_2 :_i AntStud \rightarrow (m_1 \bullet m_2) :_i Education$$

where the compound justification $m_1 \bullet m_2$ is an instance of the argument from position to know. Then, he continues by pointing towards several categories of evidence and their bibliographic references: "Evolution is well supported by evidence gathered from multiple fields of study: fossils, comparative anatomy, time and space distribution, computer simulations, and observation (2)(3)(4)(5)(6)".

$(2) :_i fossils :_i Evolution$
$(3) :_i comp\_anat :_i Evolution$
$(4) :_i time\_space\_dist :_i Evolution$
$(5) :_i simulations :_i Evolution$
$(6) :_i obs :_i Evolution$

in order to strengthen the idea that "Large amount of evidence support for evolution" ($LAEE$). A justification for it is constructed by applying the *accrual* axiom and checking the complexity of the resulting joint evidence.

$$(fossils + comp\_anat + time\_space\_dist + simulations + obs) :_i LAEE$$

, where large amount of evidence is a criterion to support evolution ($LAEE \rightarrow Evolution$). Note that the justification logic does not permit to include the evidences $(2) - (6)$ in the joint evidence, due to the right associativity of the operator (:) which gets a proof and a formula and returns a formula. The combination $(2) :_i fossils$ would not be a proper proof term of the language.

In addition, "The theory of evolution successfully predicts results in everything from fossils to psychology (9)(10)(13)." is noted as:

$$((9) + (10) + (13)) :_i fitsPrediction :_i Evolution$$

The last conveyed argument by the instigator in this round stresses the "lack of a better theory" and changes the burden of proof on the contender regarding this issue: "Can my opponent name a better theory?"

$$!^i q :_c q :_c (X \succ Evolution)$$

The link between preferred terms and preferred formulas can be:

$$(t_1 \succ t_2) \rightarrow (t_1 : F \succ t_2 : F)$$

The contender starts by clarifying that "Having evidence for something is different from convincing someone of something", denoted by

$$\neg[(t \gg_i F \rightarrow t :_i F) \wedge (t :_i F \rightarrow t \gg_i F)]$$

The justification for the above formula (refereed from now on as G) follows: "for one, they might not like what they hear and two, they might lack understanding":

$$don'tLike :_c G \vee don'tUnderstand :_c G$$

One example of attacking the arguments posted by the intrigator follows: regarding fossils, the contender considers that "fossils are facts, and they are down for interpretation like all facts are. The fossils are not evident for evolution.": $fossilsAreFacts :_c \neg fossils :_c Evolution$.

6.4 Argumentative and Explanatory Logic

We introduce here the Argument and Explanatory Logic $\mathcal{AEL}$ for differentiating argument and explanation. We also developed a computational model for cooperative labeling under the assumption of subjective views on labels.

Argument-Explanation Complementarity

The complementarity of argument and explanation in dialog should be exploited to build agents with different knowledge bases and different viewpoints that can more efficiently develop argumentation processes on their subject of interest. To model such interaction, we chose to build on the Artemov's justification logic [7].

Two individuals listening to the same debate may disagree regarding the winner of the dispute [3]. Even when they hear the same arguments and corresponding attack relations, the agents can label differently the conveyed arguments. This may be due to the fact that the situation is approached from different perspectives that reflect the capabilities and experiences of each agent, because agents care about different criteria when determining the justified conclusion [166]. A meta-level argumentation [166] is used to argue about what argument an agent should select, given a set of hierarchical structured criteria that matter for that agent. The meta argumentation viewpoint [178, 132] argues that "argumentation and dialog is necessarily a meta-logical process".

Definition 26 *An argument is a piece of reasoning $j :_i F$ in which the support j is intended by agent i to provide evidence for accepting the doubted conclusion F, as conveyed by the agent i. An explanation is a piece of reasoning $e \lhd_i F$ in which the support e (or explananum) is intended by agent i to provide a cause for the already accepted conclusion F (or explanandum).*

A_0	classical propositional axioms	
A_1	$F \rightarrow t \circ_i F$	(necessity)
A_2	$s \circ_i (F \rightarrow G) \rightarrow (t \circ_i F \rightarrow (s \cdot t) \circ_i G)$	(application)
A_4	$t \circ_i F \rightarrow !t \circ_i (t \circ_i F)$	(proof checker)
A_5	$\neg t :_i F \rightarrow ?t :_i (\neg t :_i F)$	(negative proof checker)
A_6	$t :_i F \rightarrow B_i F$	(knowledge implies belief)

Figure 6.5: Axioms of $\mathcal{AEL}$. The operator $\circ$ stands for : or $\triangleleft$.

Decision to convey argument or explanation is a process of hypothesis formation, in which the proponent develops a conjecture regarding the state of the mind of the opponent. If the proponent believes that the opponent has doubts regarding the conclusion it should convey an argument. If the proponent believes that the other party has already accepted the conclusion it can provide only an explanation that helps to augment the cognitive map of the opponent. By considering the cognitive map of the opponent the process is inherently a meta-reasoning one.

We extend the JL with explanatory capabilities, by: i) introducing the explanatory operator $t \triangleleft_i F$, where t is an explanation for F and the index i denotes the agent i providing the explanation; ii) introducing the belief operator B; iii) the possibility to interpret formulas as evidence and explanation with the conversion operator $\Downarrow$; iv) introducing labels for representing the current status of argument.

Definition 27 *The Argumentative and Explanatory Logic $\mathcal{AEL}$ contains proof terms $t \in \mathcal{T}$ and formulas $F \in \mathcal{F}$*

$$t :\ = x \mid c \mid !t \mid ?t \mid t \cdot t \mid$$
$$F :\ = p \mid F \vee F \mid \neg F \mid t :_i F \mid t \triangleleft_i F \mid B_i F \mid \Downarrow F \mid in_i(F) \mid out_i(F) \mid un_i(F)$$

Proof terms t are abstract objects that have structure. They are built up from axiom constants $c_i \in Cons$, proof variables $x, y, z, .. \in Vars$, and operators on evidence and explanations $\cdot$, !, ?. The operator precedence decreases as follows: $!, ?, \cdot, :, \triangleleft, \neg, \vee$, where $\cdot$ is left associative, and $:, \triangleleft$ right associative. The argument $t :_i F$ of agent i or its explanation $t \triangleleft_i F$ represent formulas in $\mathcal{AEL}$. To express that t is not probative evidence for agent i to support F one uses $\neg t :_i F$, respectively $\neg t \triangleleft_i F$ for non probative explanation. Parentheses are needed to express that $\neg t$ is a justification for F: $(\neg t) : F$. The evidence and explanation are used to support negated sentences too, as in $t : \neg F$ or $t \triangleleft_i \neg F$. Argumentative labels say that the formula F can be accepted by the agent i (*in*), unaccepted (*out*), or undecided yet (*un*). Similar semantics applies for the explanation operator $\triangleleft$.

The axioms of $\mathcal{AEL}$ are shown in figure 6.5, where axiom A_1 forces all formulas F to be supported by evidence or explanation. The application axiom A_2 takes

Meta Statement	Formula
Meta-argument	$j :_i (t :_i F)$
Causal argument	$j :_i (t \lhd_i F)$
Meta-explanation	$j \lhd_i (t \lhd_i F)$
Evidential explanation	$j \lhd_i (t :_i F)$
Argument-based explanation	$\Downarrow (j :_i F) \lhd_i G$
Explanation-based argument	$\Downarrow (j \lhd_i F) :_i G$

Table 6.1: Meta-argumentative semantics of $\mathcal{AEL}$.

a justifier s of an implication $F \rightarrow G$ and a justifier t of its antecedent F, and produces a justification $s \cdot t$ of the consequent G. Differently from the classical definition of an abstract argument, where the support represents a set which is minimal and without structure, here the support t represents an explicit proof term facilitating access to the reasoning chain of the agent conveying the argument.

Example 30 *Bird is the justification of agent i for the sentence Fly, given by $bird :_i Fly$. The penguins, which are birds ($penguin \rightarrow bird$), represent an exception, which according to agent j, blocks the acceptability of evidence bird as being enough for the sentence Fly. The application operator is used to model the exception: $[penguin \cdot (penguin \rightarrow bird)] :_j \neg bird :_i Fly$.*

Arguments and explanations are assumed to be verified. The operator ! represents a request for a positive proof, while the negative proof checker ? forces agents to provide evidence why they are not able to justify a particular formula F. Thus $!t :_i G$ represents a request for evidence, while $!e \lhd_i G$ a request for explanation. A common usage of these operators occurs in judicial cases, where "evidence for" coexists with "explanation against“ or ”lack of evidence against” coexists with ”explanation for”. The axiom A_6 encapsulates the classical relation between knowledge an belief, with the difference that in our case knowledge is explicitly encapsulated in the proof term t.

The meta-argumentative semantics of $\mathcal{AEL}$ (table 6.1) is given by the constraint imposed by axiom A_1: the argument $t :_i F$ or the explanation $t \lhd_i F$ represent formulas, which should have their own justification terms. This corresponds to the principle of inferential justification: for sentence F to be justified on the basis of t one must justify that t makes F plausible. Given the right associativity of :, the term j in $j :_i t :_i F$ represents a statement about an argument, defined as a *meta-argument* in [178]. Constants are used to stop the ad infinitum meta-argumentation chain by representing a kind of justification that does not depend on other justifiers. Arguments with causal statements in their conclusions are

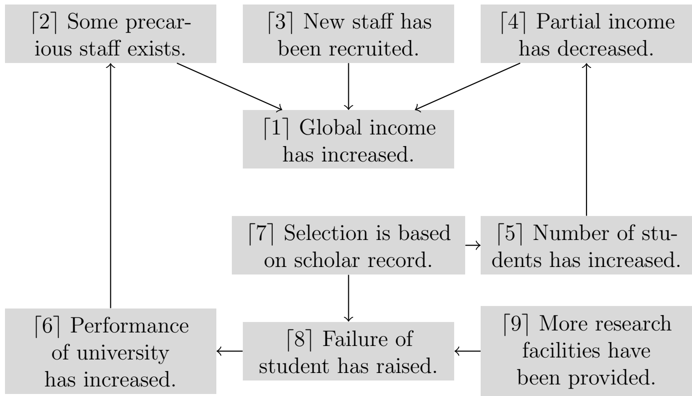


Figure 6.6: Argumentation framework for the running scenario.

called *causal arguments* in [127]. In $j :_i \neg(e \lhd_k F)$, the agent i constructs a causal argument attacking the explanation e provided by the agent k for the statement F. In case of *meta-explanation* $j \lhd_i : (t \lhd_i F)$, an explanation j is provided why t is a cause for F. An *evidential explanation* $j \lhd_i : (t :_i F)$ identifies a cause j for the argument $t :_i F$. The expressivity of $\mathcal{AEL}$ allows agent i to request a causal argument to agent j $(!t :_i e \lhd_j F)$, request a meta-argument $(!t :_i e :_j F)$, a meta-explanation $(!t \lhd_i e \lhd_j F)$ or an evidential explanation $(!t \lhd_i e :_j F)$. Introspection occurs when the agent i is the same as the agent j.

Definition 28 *We say that a formula F attacks another formula G according to agent i, if F acts as a justification for $\neg G$, given by $\Downarrow F :_i \neg G$, meaning that the bounded rational agent i which accepts F would have to reject G.*

To model the distinction between argument and explanation we use the labeling approach from argumentation theory [23], with the argumentation framework $\delta = (\Delta, att)$ and the total labeling function L for agent i $L : \Delta \times i \to \{in,\ out,\ un\}$.

Definition 29 *A complete labeling is a labeling such that for every $t \in \Delta$ it holds that: i) if t is labelled "in" then all attackers of t are labelled "out"; ii) if all attackers of t are labelled "out" then t is labelled "in"; iii) if t is labelled "out" then t has an attacker that is labelled "in"; and iv) if t has an attacker that is labelled "in" then t is labelled "out".*

Table 6.2: Agent labeling functions: $\mathcal{AA}$ stands for agent a own labels, whilst $\mathcal{AB}$ for agent a subjective view on b' labeling.

	Notation	$\lceil 1 \rceil$	$\lceil 2 \rceil$	$\lceil 3 \rceil$	$\lceil 4 \rceil$	$\lceil 5 \rceil$	$\lceil 6 \rceil$	$\lceil 7 \rceil$	$\lceil 8 \rceil$	$\lceil 9 \rceil$
$\mathcal{L}(\Delta, a)$	$\mathcal{AA}$	out	out	in	in	out	in	in	out	out
$B_a\mathcal{L}(\Delta, b)$	$\mathcal{AB}$	in	out	out	out	in	out	out	out	in
$\mathcal{L}(\Delta, b)$	$\mathcal{BB}$	out	in	un	out	in	out	out	in	out
$B_b\mathcal{L}(\Delta, a)$	$\mathcal{BA}$	out	out	in	un	un	in	un	out	in

6.5 Arguing based on subjective views

Initial State

Let the case in Fig. 6.6, where the argument “new staff has been recruited“ ($\lceil 3 \rceil$) attacks the argument ”global income has increased” ($\lceil 1 \rceil$), represented by $\Downarrow \lceil 3 \rceil :_{\{a,b\}} \lceil 1 \rceil$. At this initial state, both the proponent agent a and the opponent agent b accept the existence of an attack relation between $\lceil 3 \rceil$ and $\lceil 1 \rceil$.

Under the same assumption for all arguments, δ is considered common knowledge for the agents, with difference in how they label the arguments. Assuming the complete labelings in table 6.2, the first line represents the labeling function $\mathcal{L}(\Delta, a)$ of agent a for each topic in Δ, and the second line represents the beliefs of a on the labeled function of agent b. The shortcut $B_a\mathcal{L}(\Delta, b)$ is used to represent the belief set $B_a in_b(\lceil 1 \rceil) \wedge B_a out_b(\lceil 2 \rceil) \wedge_a out_b(\lceil 3 \rceil) \wedge ... \wedge B_a in_b(\lceil 9 \rceil)$ for each argument $\lceil t \rceil \in \Delta$. The graphical representation of each agent perspectives on Δ is shown in Fig. 6.7. Note that all the labels follow the constraints in definition 29.

Computing agreements and disagreements

Given the above input, the agents proceed to identify current agreements and disagreements or possible agreements or disagreements, with the algebra in figure 6.8. The four worlds are considered relevant here (table 6.3). The actual world w_O identifies conflicts and agreements based on the current labels of each agent $\mathcal{L}(\Delta, a)$, $\mathcal{L}(\Delta, b)$. The world w_a perceived by agent a defines conflicts and agreements on the a labels $\mathcal{L}(\Delta, a)$ and its initial beliefs about b's labels $B_a\mathcal{L}(\Delta, b)$, and similarly for the world w_b perceived by b. The subjective world w_S is constructed based on the subjective views of the agents.

Definition 30 *The lower bound subjective agreement* $\underline{SA}_{xy}$ *of agent x regarding agent y represents the set of concepts having the same labels ”in” or ”out” according to agent x perspective on agent y:* $\underline{SA}_{xy} = \{t \mid \mathcal{XX}(t) = \mathcal{XY}(t) = in$ *or*

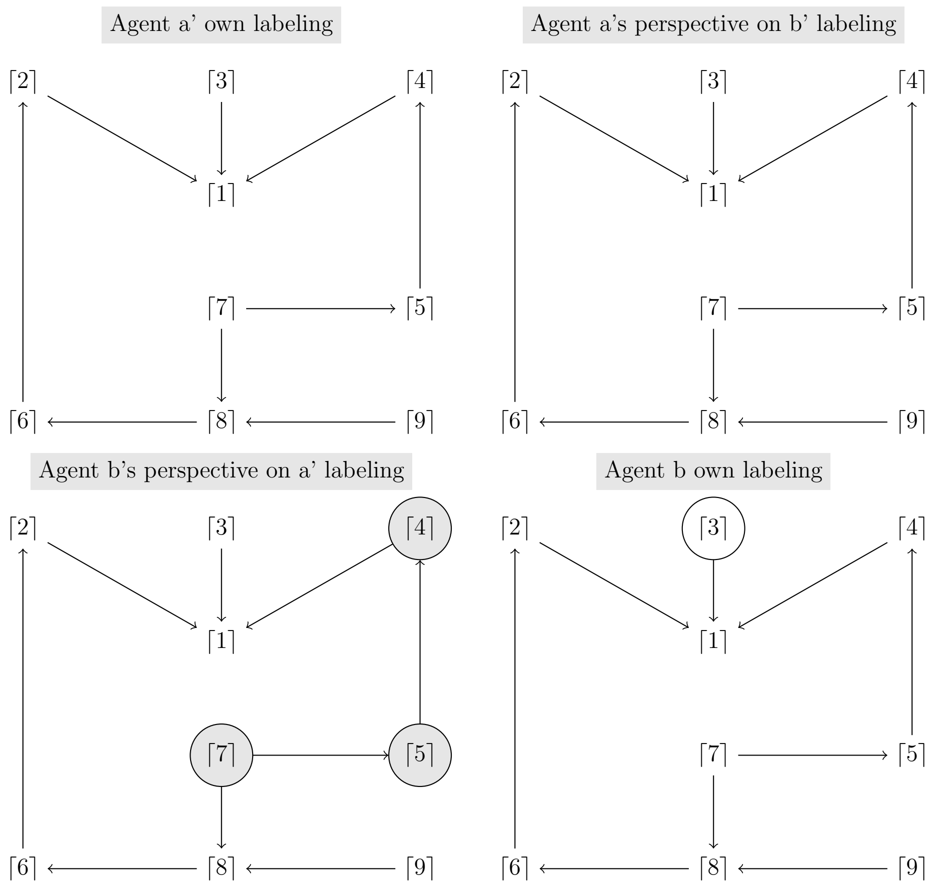


Figure 6.7: Subjective views of the agents: grey boxes represent arguments labelled *in*, white boxes *out*, whilst circle *un*.

out + out = $\oplus$	in + out = $\ominus$	un + out = $\odot$
out + in = $\ominus$	in + in = $\oplus$	un + in = $\odot$
out + un = $\odot$	in + un = $\odot$	un + un = $\odot$

Figure 6.8: Labeling algebra: $\oplus$ means agreement, $\ominus$ disagreement, $\odot$ undecided.

Table 6.3: Worlds of labels. $\oplus$ stands for agreement, $\ominus$ for disagreement, $\odot$ undecided yet. w_O is the actual world, w_a is agent'a world, w_b agent's b world, and w_S is the subjective world.

	World labels	$\lceil 1 \rceil$	$\lceil 2 \rceil$	$\lceil 3 \rceil$	$\lceil 4 \rceil$	$\lceil 5 \rceil$	$\lceil 6 \rceil$	$\lceil 7 \rceil$	$\lceil 8 \rceil$	$\lceil 9 \rceil$
w_O	$\mathcal{AA}+\mathcal{BB}$	$\oplus$	$\ominus$	$\odot$	$\ominus$	$\ominus$	$\ominus$	$\ominus$	$\ominus$	$\oplus$
w_a	$\mathcal{AA}+\mathcal{AB}$	$\ominus$	$\oplus$	$\ominus$	$\ominus$	$\ominus$	$\ominus$	$\ominus$	$\oplus$	$\ominus$
w_b	$\mathcal{BB}+\mathcal{BA}$	$\oplus$	$\ominus$	$\odot$	$\odot$	$\odot$	$\ominus$	$\odot$	$\ominus$	$\ominus$
w_S	$\mathcal{AB}+\mathcal{BA}$	$\ominus$	$\oplus$	$\ominus$	$\odot$	$\odot$	$\ominus$	$\odot$	$\oplus$	$\oplus$

$\mathcal{XX}(t) = \mathcal{XY}(t) = out\}$. The upper bound subjective agreement $\overline{SA}_{xy}$ supplementary includes the topics labelled "un" by one agent: $\overline{SA}_{xy} = \underline{SA}_{\mathcal{XY}} \cup \{t \mid \mathcal{XX}(t) = UNor\mathcal{XY}(t) = UN\}$. The lower bound subjective disagreement set $\underline{SD}_{xy}$ of agent x towards y represents the arguments having different labels "in" or "out" according to agent x view on agent y. The upper bound subjective disagreement set $\overline{SD}_{xy}$ additionally includes the topics labelled "un" by one agent.

Using the operators in figure 6.8, $\underline{SA}_{ab} = \{t \mid w_a = \oplus\} = \{\lceil 2 \rceil, \lceil 8 \rceil\}$. No indeterminacy existing in w_a, the upper bound set $\overline{SA}_{ab}$ does not include any extra argument, given by $\overline{SA}_{ab} = \{t \mid w_a = \oplus \vee \odot\} = \{\lceil 2 \rceil, \lceil 8 \rceil\}$. From b's perspective, $\underline{SA}_{ba} = \{t \mid w_b(t) = \oplus\} = \{\lceil 1 \rceil\}$, whilst $\overline{SA}_{BA} = \{t \mid w_b(t) = \oplus \vee \odot\} = \{\lceil 1 \rceil, \lceil 3 \rceil, \lceil 4 \rceil, \lceil 5 \rceil, \lceil 7 \rceil\}$. The subjective disagreements according to the world w_a of agent a is $\underline{SD}_{ab} = \{t \mid w_a(t) = \ominus\} = \overline{SD}_{ab} = \{\lceil 1 \rceil, \lceil 3 \rceil, \lceil 4 \rceil, \lceil 5 \rceil, \lceil 6 \rceil, \lceil 7 \rceil, \lceil 9 \rceil\}$. Observe that the upper bound disagreements $\overline{SD}_{ab} = \{t \mid w_a(t) = \ominus \vee \odot\} = \underline{SD}_{ab}$. From its partner perspective, the disagreement looks like $\underline{SD}_{ba} = \{t \mid w_b(t) = \ominus\} = \{2, 6, 8, 9\}$, respectively the upper bound disagreement $\overline{SD}_{ba} = \underline{SD}_{ba} \cup \{\lceil 3 \rceil, \lceil 4 \rceil, \lceil 5 \rceil, \lceil 7 \rceil\}$.

Containing agreed conclusions, the set $\underline{SA}_{ab}$ represents the topics on which a is expecting only explanations from its partner (Table 6.4). By including only disagreed conclusions, the set $\underline{SD}_{ab}$ contains topics on which agent a is expecting arguments only. For the elements in $\overline{SD}_{ab} \setminus \underline{SD}_{ab}$, agent a expects hearing or may convey both explanations and arguments. For agent b, arguments in $\overline{SA}_{ba} \setminus \underline{SA}_{ba}$ both evidential explanations or meta-arguments are expected. By addressing topics in $\overline{SA}_{ba} \setminus \underline{SA}_{ba}$, b tries to further identify possible agreements. By explaining topics in $\underline{SA}_{ba}$, b tries to extend the cognitive map of a by providing its explanations on agreed labels. By discussing the arguments in $\overline{SD}_{ba}$, agent b tries to solve the conflict as defined according to its view. While an agent believes that it has conveyed an argument or an explanation, in fact it has not. The rightness on conveying either argument or explanation should be computed based on the objective world w_O.

Table 6.4: Expecting arguments or explanations.

Expect	$w_x(t)$	Agent a	Agent b
Exp. only	$\oplus$	$\underline{SA}_{ab} = \{\lceil 2\rceil, \lceil 8\rceil\}$	$\underline{SA}_{ba} = \{\lceil 1\rceil\}$
Arg. only	$\ominus$	$\underline{SD}_{ab} = \{\lceil 1\rceil, \lceil 3\rceil, \lceil 4\rceil, \lceil 5\rceil, \lceil 6\rceil, \lceil 7\rceil\ \lceil 9\rceil\}$	$\underline{SD}_{ba} = \{\lceil 2\rceil, \lceil 6\rceil, \lceil 8\rceil, \lceil 9\rceil\}$
Both	$\odot$	$\overline{SA}_{ab} \cap \overline{SD}_{ab} = \{\}$	$\overline{SA}_{ba} \cap \overline{SD}_{ba} = \{\lceil 3\rceil, \lceil 4\rceil, \lceil 5\rceil, \lceil 7\rceil\}$

$\ominus_O + \ominus_x = \ominus_x^w$	conflict rightness	$\odot_O + \oplus_x = \odot_x^{\neg w}$	inadvertence not aware
$\oplus_O + \oplus_x = \oplus_x^w$	agreement rightness	$\odot_O + \ominus_x = \odot_x^{\neg w}$	inadvertence not aware
$\odot_O + \odot_x = \odot_x^w$	inadvertence rightness	$\ominus_O + \odot_x = \circledast_x^w$	possible conflict rightness
$\oplus_O + \ominus_x = \oplus_x^{\neg w}$	agreement not aware	$\oplus_O + \odot_x = \circledcirc_x^w$	possible agreement rightness
$\ominus_O + \oplus_x = \ominus_x^{\neg w}$	conflict not aware		

Figure 6.9: Rightness/inadvertence on expecting/conveying argument or explanation. First term represents the actual world w_O, the second term is the subjective perspective of agent x.

Adequacy of conveying/expecting argument or explanation

Given the difference between expecting explanations or arguments (subjective worlds w_a and w_b) and objective world w_O, the agents may wrongly expect explanations instead of arguments and vice-versa. For the rightness or adequacy of conveying/expecting argument or explanation, the algebra in Fig. 6.9 is used.

Definition 31 *The lower bound objective agreement $\underline{OA}$ represents the set of concepts having the same labels "in" or "out" according to agents own labelings $\underline{OA} = \{t \mid \mathcal{AA}(t) = \mathcal{BB}(t) = in \text{ or } \mathcal{AA}(t) = \mathcal{BB}(t) = out\}$. The upper bound objective agreement $\overline{OA}$ supplementary includes the topics labelled "un" by one agent: $\overline{OA} = \underline{OA} \cup \{t \mid \mathcal{AA}(t) = un \text{ or } \mathcal{BB}(t) = un\}$. The lower bound objective disagreement $\underline{OD}$ includes the topics which are labelled differently "in" or "out", given by $\underline{OD} = \Delta \setminus \overline{OA}$. The upper bound objective disagreement also includes the topics which are undecided by one party, given by $\overline{OD} = \Delta \setminus \underline{OA}$*

The topics $t \in \Delta$ for each agent a is right on the agreement form the set of adequate explanations for a: $\underline{OA} = \{t \mid w_O(t) = \oplus\} = \{\lceil 1\rceil, \lceil 9\rceil\}$ and $\overline{OA} = \{t \mid w_O(t) = \oplus \vee \odot\} = \{\lceil 1\rceil, \lceil 9\rceil, \lceil 3\rceil\}$ (line 1 in table 6.3). Based on line 1 in table 6.3, $\underline{OD} = \{t \mid w_O(t) = \ominus\} = \{\lceil 2\rceil, \lceil 4\rceil, \lceil 5\rceil, \lceil 6\rceil, \lceil 7\rceil, \lceil 8\rceil\}$. $\overline{OD}$ additionally includes topic $\lceil 3\rceil$ which may introduce disagreement in the light of new information. $\overline{OA}$ includes the topics for each would be legitimate to provide explanations. $\overline{OD}$ contains the topics for each would be legitimate to provide arguments.

Definition 32 *The set of adequate explanations $\underline{AE}$ for an agent x represents the lower bound agreements on which x is right ($\oplus_x^w$), given by $\underline{AE}_{x^w} = \underline{OA} \cap \underline{SA}_{xy}$. The*

Table 6.5: Agreement and conflict awareness for agents a and b. $\oplus^{w}$ stands for agreement awareness, $\oplus^{\neg w}$ stands for agreement ignorance, $\ominus^{w}$ for disagreement awareness, $\ominus^{\neg w}$ for disagreement ignorance, $\odot^{w}$ for ignorance awareness, $\odot^{\neg w}$ for not aware of its own ignorance.

Awareness and ignorance	$\lceil 1 \rceil$	$\lceil 2 \rceil$	$\lceil 3 \rceil$	$\lceil 4 \rceil$	$\lceil 5 \rceil$	$\lceil 6 \rceil$	$\lceil 7 \rceil$	$\lceil 8 \rceil$	$\lceil 9 \rceil$
Agent a: $w_a + w_O$	$\oplus_a^{\neg w}$	$\ominus_a^{\neg w}$	$\odot_a^{\neg w}$	$\ominus_a^{w}$	$\ominus_a^{w}$	$\ominus_a^{w}$	$\ominus_a^{w}$	$\ominus_a^{\neg w}$	$\oplus_a^{\neg w}$
Agent b: $w_b + w_O$	$\oplus_b^{w}$	$\ominus_b^{w}$	$\odot_b^{w}$	$\circledast_b^{w}$	$\circledast_b^{w}$	$\ominus_b^{w}$	$\circledast_b^{w}$	$\ominus_b^{w}$	$\oplus_b^{\neg w}$

set of possible adequate explanations $\overline{AE}$ for an agent x is given by the upper bound agreements on which agent x is right ($\odot_x^{w}$), computed by $\overline{AE}_{x^w} = \overline{OA} \cap \overline{SA}_{xy}$. The set of inadequate explanations IE for an agent x contains the topics for which x has not identified a conflict between labels ($\ominus_x^{\neg w}$ or $\circledast_x^{w}$), given by $IE_{x^{\neg w}} = \underline{OA} \setminus \underline{SA}_{xy}$.

For each $t \in \Delta$, $\underline{A}_{a^w} = \{t \mid [w_O + w_a](t) = \oplus_a^{w}\} = \{\lceil 1 \rceil, \lceil 9 \rceil\} \cap \{\lceil 2 \rceil, \lceil 8 \rceil\} = \varnothing$, whilst $\overline{A}_{a^w} == \{t \mid [w_O + w_a](t) = \oplus_a^{w} \vee \odot_a^{w}\} = \{\lceil 1 \rceil, \lceil 9 \rceil, \lceil 3 \rceil\} \cap \{\lceil 2 \rceil, \lceil 8 \rceil\} = \varnothing$. The agreement rightness for agent b, $\underline{A}_{b^w} = \{\lceil 1 \rceil, \lceil 9 \rceil\} \cap \{\lceil 1 \rceil\} = \{\lceil 1 \rceil\}$ represents the only topic on which agent b, if he has decided to convey an explanation, that explanation would be adequate in the objective world w_O.

Definition 33 *The set of adequate arguments $\underline{AA}$ for an agent x represents the lower bound disagreements on which x is right ($\ominus_x^{w}$), given by $\underline{AA}_x^{\neg w} = \underline{OD} \cap \underline{SD}_{xy}$. The set of possible adequate arguments $\overline{AA}$ for an agent x is given by the upper bound disagreements on which agent x is right ($\odot_x^{w}$), computed by $\overline{A}_{x^w} = \overline{OD} \cap \overline{SD}_{xy}$. The set of inadequate arguments for an agent x contains the topics for which x is not aware of an agreement between labels ($\oplus_x^{\neg w}$), given by $\underline{A}_{x^{\neg w}} = \underline{OA} \setminus \underline{SA}_{xy}$.*

Agent a is not aware that it shares the same labels with agent b regarding topics $\overline{A}_{a^{\neg w}} = \{t \mid [w_O + w_b](t) = \oplus_a^{\neg w}\} = \{\lceil 1 \rceil,\ \lceil 9 \rceil\}$, so it will wrongly convey arguments instead of explanations (not adequate a' arguments in table 6.6). At the same time, b is not aware $\overline{A}_{b^{\neg w}}$ that an agreement exists on topic $\lceil 9 \rceil$.

The results in table 6.5 are derived by reporting the agent a world w_a to the objective world w_O, respectively the agent b world to the same objective world w_O. Not being aware that an agreement exists on topic $\lceil 1 \rceil$, a is not expecting explanations and also it will not convey explanations, but only arguments, on the topic $\lceil 1 \rceil$. Instead, b has a correct cognitive representation about the agreement on topic $\lceil 1 \rceil$. Not being aware about the conflict on topic $\lceil 2 \rceil$, a will wrongly utter explanations instead of arguments. Being right on this conflict, b will correctly convey arguments and not explanations. Agent a is not aware that a possible

Move	Adequacy	Op	Agent a	Agent b
Exp.	adequate	$\oplus_x^w$		$\lceil 1 \rceil$
	$\neg$ adequate	$\ominus_x^{\neg w} \vee \circledast_x^w$	$\lceil 2 \rceil$, $\lceil 8 \rceil$	$\lceil 4 \rceil$, $\lceil 5 \rceil$, $\lceil 7 \rceil$
	possible	$\odot_x^w \vee \odot_x^{\neg w}$		$\lceil 3 \rceil$
Arg.	adequate	$\ominus_x^w \vee \circledast_x^w$	$\lceil 4 \rceil$, $\lceil 5 \rceil$, $\lceil 6 \rceil$, $\lceil 7 \rceil$	$\lceil 2 \rceil$, $\lceil 6 \rceil$, $\lceil 8 \rceil$, $\lceil 4 \rceil$, $\lceil 5 \rceil$, $\lceil 7 \rceil$
	$\neg$ adequate	$\oplus_x^{\neg w}$	$\lceil 1 \rceil$, $\lceil 9 \rceil$	$\lceil 9 \rceil$
	possible	$\odot_x^w \vee \odot_x^{\neg w}$	$\lceil 3 \rceil$	$\lceil 3 \rceil$

Table 6.6: The adequacy of using arguments (:) or explanations ($\triangleleft$).

agreement exists on topic $\lceil 3 \rceil$. Having its own label undecided yet on topic $\lceil 3 \rceil$, given by $un_b(\lceil 3 \rceil)$, agent b is obviously aware that a possible labeling conflict may occur during the debate.

Therefore, if an agent decides to utter an explanation or argument it may be wrong or right depending on the combination between w_O, w_a and w_b (table 6.6). According to its cognitive map, a tends to provide explanans for topics $\lceil 2 \rceil$ and $\lceil 8 \rceil$ (table 6.4). Uttering an explanation is not adequate in both cases due to the existence of a conflict in w_O, given by $w_O(\lceil 2 \rceil) = w_O(\lceil 8 \rceil) = \ominus$. From the set of possible argumentative moves of a, an argument supporting the topic $\lceil 1 \rceil$) would be inadequate because there is agreement on the labels in the actual world w_O, given by $w_O(\lceil 2 \rceil) = \oplus$. The argument on topic $\lceil 3 \rceil$) is possible to be an adequate argument for the, b which at the moment is not decided with respect to label of $\lceil 3 \rceil$). Each topic for both expectation and argument can be conveyed according to its representation (last line in table 6.4): appears once as argument and once as an explanation. Each such occurrence is categorised as adequate, inadequate or possible adequate in table 6.6. For instance, an explanation would be inadequate for topics $\lceil 4 \rceil$) and $\lceil 5 \rceil$), but arguments would be adequate. The topic $\lceil 3 \rceil$) is the only one adequate to be both explained or argued, due to its undecided status in the objective world w_O.

6.6 Updating labels based on move adequacy

Dialog strategy

The dialog strategy of an agent consists of interleaving argumentation games (:) with explanatory games ($\triangleleft$). For the argumentative part (:) an agent can choose between requesting a positive proof (!), a negative one (?), or providing an argument. Both the request and the provided argument regard the labels *"un"*, *"in"*, and *"out"*. For the explanatory part ($\triangleleft$) an agent can choose between a positive proof of the explanandum or for providing an explanation, regarding one of the three labels *"un"*, *"in"*, and *"out"*. Depending on the way of traversing the tree, different strategies may be defined. A possible strategy would have the following steps: 1) obtain explanations regarding unlabelled arguments; 2) provide explanations regarding unlabelled arguments; 3) obtain explanations regarding arguments with the same labels; 4) provide explanations regarding arguments with the same labels; 5) request arguments regarding arguments with different labels; 6) provide arguments regarding arguments with different labels. The strategy aims to clarify the undecided topics (steps 1 and 2), then it tries to extend to cognitive map of each agent by focusing on the subjective perceived as agreed arguments (steps 3 and 4), and finally it deals with the subjective perceived as conflicting labels. In this strategy the agent prefers to obtain information first and after that to convey his own arguments or explanations.

The strategy is defined based on information in table 6.4, where the computation assumes that agents have access to their own worlds only w_x and w_y. The algorithm gets as input the current Δ, the labeling function of the agent to move $\mathcal{L}(\Delta, x)$, and its initial perspective $B_x\mathcal{L}(\Delta, y)$ on agent y and it returns to the next move. The strategy commences by clarifying the topics where the agent x is not sure that an agreement or conflict exists. Assuming that it is the turn of b, then it has to clarify a topic from $\overline{SD}_{\mathcal{BA}} \setminus \underline{SD}_{\mathcal{BA}} = \{3, 4, 5, 7\}$. From the selected topic t, the agent checks the source of undecidability. If it is due to its own labeling function $un_x(t)$ he has to introspect its own knowledge base. If it is not able to find an adequate justification either for *"in"* or *"out"*, it accepts the label proposed by its partner. In case this is *"un"* too, it selects the next topic. If no topic exists it requests for justification trying to force agent y to label differently. Otherwise, if label is *"in"* or *"out"*, the indeterminacy comes from the other party, thus agent x has to provide its own positive justification for the current label.

Case analysis

Expecting argument, receiving argument. For instance, topic $\lceil 3 \rceil$ lies in this case, which is a possible adequate argument in w_o for both agents a and b. Being $un_b(\lceil 3 \rceil)$, agent b can provide evidence t supporting the current undecided label: $t :_b un_b(\lceil 3 \rceil)$. Receiving what is expecting, the agent's a beliefs $B_a(L_b, \Delta)$ are not attacked, thus it does not have to adjust its cognitive map $\mathcal{AB}$. Agent a replies with an argument supporting its label $t' :_a in_a\lceil 3 \rceil$. Note that agent a is not in a position to convey explanations on $\lceil 3 \rceil$ according to table 6.4. Agent b is expecting both arguments or explanations on $\lceil 3 \rceil$. Receiving the argument $t' :_a in_a\lceil 3 \rceil$, it also does not have to adjust its representation $\mathcal{BA}$ about a. By accepting the argument, b's own labels $\mathcal{BB}$ are affected, conflicts and agreements are updated and the strategy algorithm selects a new move for the current situation.

Expecting explanation, receiving argument. Agent a expects arguments regarding topic $\lceil 2 \rceil$, whilst agent b conveys only arguments on $\lceil 2 \rceil$. Note that agent a has a wrong representation on $\lceil 2 \rceil$, identified in table 6.6 as inadequate explanation. Receiving an argument $u :_b in_b(\lceil 2 \rceil)$, this is enough evidence for agent a to update its representation $\mathcal{AB}$ on agent b, given by $[u :_b in_b(\lceil 2 \rceil)] :_b in_b(\lceil 2 \rceil)$ and based on axiom A_6 follows that $B_a in_b(\lceil 2 \rceil)$. Consequently, a new disagreement has been identified, which triggers new computations.

Expecting argument, receiving explanation. Consider that b provides an explanation $e \lhd_b \lceil 1 \rceil$. Agent a identifies a conflict in its map $\mathcal{AB}$, in which the objective agreement on $\lceil 1 \rceil$ was treated as a disagreement. Observe also that if b had decided to explain the argument $\lceil 3 \rceil$ instead of arguing on it, the agent a would have been able to identify the objective agreement on the focal topic $\lceil 3 \rceil$.

6.7 Conclusions

Contributions in this chapter consist of: i) proposing the $\mathcal{AEL}$ for distinguishing between argument and explanation. ii) developing a computational model for cooperative labeling when agents have subjective views on labels. Quite aware of the difficulty of formalising and applying meta-argumentation, we have embarked on this task aiming to facilitate agent understanding in guiding the dialog between them. A research direction would be to use proof nets for visualising argumentation in the justification based logic $\mathcal{AEL}$.

We considered the complementarity of argument and explanation in dialog, aiming to model the interaction of agents with different knowledge bases and different viewpoints. The Justification Logic was extended with arguments and explanations, resulting in a new logic called Argument and Explanatory Logic ($\mathcal{AEL}$). $\mathcal{AEL}$ provides the means to better use agents complementary knowledge on the subject being discussed. The $\mathcal{AEL}$ was applied on a cooperative labeling argumentation by agents with different views.

Chapter 7

Arguing in hybrid logics

> ”It is easier to change the specification to fit the program than vice versa“
>
> Alan J. Perlis

7.1 Motivation

We illustrate here a recommender framework for assisting flight controllers. The system combines argumentation theory and model checking in the evaluation of trade-offs to be made in the presence of incomplete and potentially inconsistent information. We view a Hybrid Kripke model as a description of an air traffic control (ATC) domain. We apply a decision strategy based on Hybrid Logics and Defeasible Reasoning to assist the process of model update when the system has to accommodate new properties or norm constraints. When the model fails to verify a property, a defeasible logic program is used to analyze the current state and to apply updating operations on the model.

Our hypothesis is that argumentation can be used to assure safety in complex critical systems by providing a way of assisting end-users to reach rationally justified decisions. Landing criteria for assuring safety in complex landing situations are modeled as a DeLP program. Prospective decisions are presented to the system as queries. Given a query representing a decision concerning a safety requirement w.r.t. such a set of criteria, the DeLP engine will engage in an introspective dialectical process considering pros and cons against a decision and will answer a recommendation in the case that there is a warrant for the query. Besides, as in a real-time environment in which border conditions may vary from second to second, decisions cannot be taken with respect to a static DeLP program. Thus, we present a framework for making recommendations based on sensor input regarding the values of the parameters characterizing the safety problem.

7.2 Model Repair for an Unmanned Aircraft Vehicle

Given a Kripke structure $\mathcal{M}$ and a formula ϕ, with $\mathcal{M}\neg \models \phi$, the task of *model repair* is to obtain a new model $\mathcal{M}'$ such that $\mathcal{M}' \models \phi$. We consider the following primitive update operations [182].

Definition 34 *Given $\mathcal{M} = (S, R, L)$, the updated model $\mathcal{M} = (S', R', L')$ is obtained from $\mathcal{M}$ by applying the* primitive update operations*:*

1. *(PU_1) Adding one relation element: $S' = S$, $L' = L$, and $R' = R \cup \{(s_i, s_j)\}$ where $(s_i, s_j) \notin R$ for two states $s_i, s_j \in S$.*
2. *(PU_2) Removing one relation element: $S' = S$, $L' = L$, and $R' = R \backslash \{(s_i, s_j)\}$ where $(s_i, s_j) \notin R$ for two states $s_i, s_j \in S$.*
3. *(PU_3) Changing labeling function in one state: $S' = S$, $R' = R$, $s^* \in S, L'(s^*) \neq L(s^*)$, and $L'(s) = L(s)$ for all states $s \in S \setminus \{s^*\}$.*
4. *(PU_4) Adding one state: $S' = S \cup \{s^*\}, s \notin S$, $R' = R$, $\forall\, s \in S, L'(s) = L(s)$.*

Our task is to build an argumentative based decision procedure that takes as input a model $\mathcal{M}$ and a formula ϕ, it outputs a model $\mathcal{M}'$ where ϕ is satisfied. Figure 7.1 depicts the proposed *model repair* framework.

The task addressed here focuses on a situation on which the specification of the model is not consistent. Consider the following two "rules of the air" [175]:

R_3: *Collision Avoidance – "When two UAVs are approaching each other and there is a danger of collision, each shall change its course by turning to the right."*

R_4: *Navigation in Aerodrome Airspace – "An unmanned aerial vehicle passing through an aerodrome airspace must make all turns to the left [unless told otherwise]."*

Let

$$\mathcal{A}_2 = \left\{ \begin{array}{l} \mathit{alter\_course}(\mathit{uav_1}, \mathit{right}) \prec \mathit{aircraft}(\mathit{uav_1}), \mathit{aircraft}(\mathit{uav_2}) \\ \qquad \mathit{collision\_hazard}(\mathit{uav_1}, \mathit{uav_2}) \\ \mathit{collision\_hazard}(\mathit{uav_1}, \mathit{uav_2}) \prec \mathit{approaching\_head\_on}(\mathit{uav_1}, \mathit{uav_2}), \\ \qquad \mathit{distance}(\mathit{uav_1}, \mathit{uav_2}, X), X < \mathit{1000} \end{array} \right\}$$

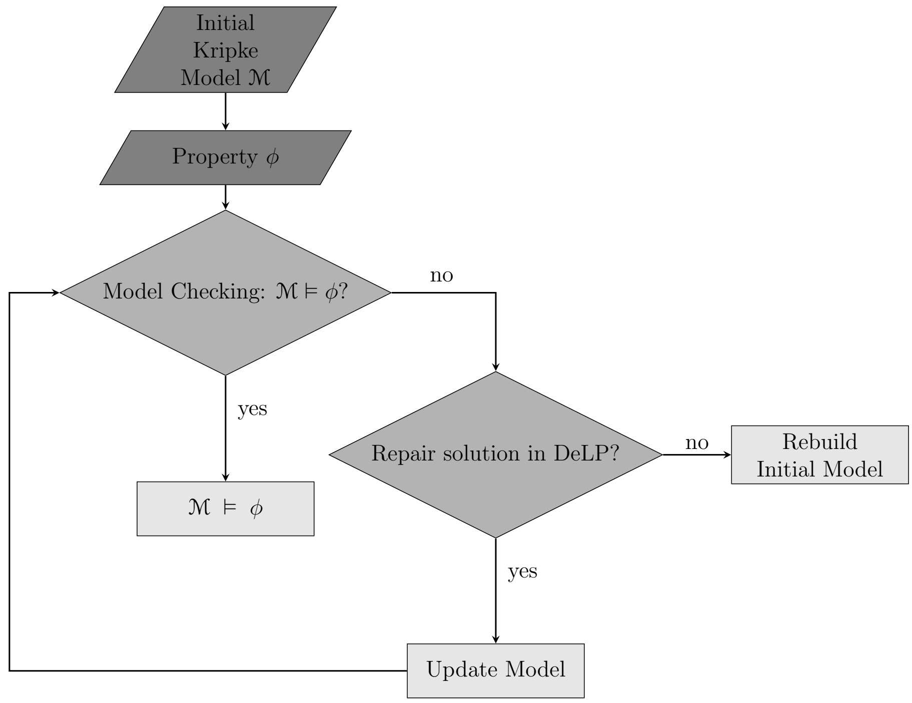


Figure 7.1: Argumentative model repair algorithm

in the argument $\langle \mathcal{A}_2, \mathit{alter\_course}(\mathit{uav_1}, \mathit{right}) \rangle$, a collision hazard occurs when two aerial vehicles uav_1 and uav_2 approach head on, and the distance between them is smaller than a threshold. The collision hazard further triggers the necessity to alter the course to the right, according to the R_3 specification. Let

$$\mathcal{A}_3 = \left\{ \begin{array}{l} \mathit{alter\_course}(\mathit{uav_1}, \mathit{left}) \prec \mathit{aircraft}(\mathit{uav_1}), \mathit{nearby}(\mathit{uav_1}, \mathit{aerodrom}), \\ \qquad \mathit{change\_direction\_required}(\mathit{uav_1}) \\ \mathit{change\_direction\_required}(\mathit{uav_1}) \prec \mathit{collision\_hazard}(\mathit{uav_1}, \mathit{uav_2}) \end{array} \right\}$$

in the argument $\langle \mathcal{A}_3, \mathit{alter\_course}(\mathit{uav_1}, \mathit{left}) \rangle$, if a change of direction is required in the aerodrome airspace, the direction should be altered to the left. A possible conflict occurs between arguments $\langle \mathcal{A}_2, \mathit{alter\_course}(\mathit{uav_1}, \mathit{right}) \rangle$ and $\langle \mathcal{A}_4, \sim \mathit{alter\_course}(\mathit{uav_1}, \mathit{right}) \rangle$ where:

$$\mathcal{A}_4 = \left\{ \sim \mathit{alter\_course}(\mathit{uav_1}, \mathit{right}) \prec \mathit{alter\_course}(\mathit{uav_1}, \mathit{left}) \right\}.$$

The command $\langle \mathcal{A}_5, \sim \mathit{alter\_course}(\mathit{uav_1}, \mathit{left}) \rangle$ conveyed from the ground control system to change direction to the right acts as a defeater for the argument $\mathcal{A}_3$, where (notice that strict rules should not form part of argument structures as they are not points of attack, we abuse notation here just for emphasis):

$$\mathcal{A}_5 = \left\{ \sim \mathit{alter\_course}(\mathit{uav_1}, \mathit{left}) \leftarrow \mathit{conveyed\_command\_course}(\mathit{uav_1}, \mathit{right}) \right\}$$

Assume that the current model $\mathcal{M}$ satisfies the specification R_3. The task is to repair $\mathcal{M}$ with $\mathcal{M}'$ which also satisfies R_4. Our solution starts by treating rules R_3 and R_4 as arguments. The conflict between them are solved by a defeasible theory encapsulated as DeLP program, which outputs a dialectical tree of the argumentation process. The information from this tree is further exploited to decide which primitive update operations PU_i are required to repair the model.

7.3 Interleaving Arguing and Model Checking

We propose an automated solution for model self-adaption based on model checking in Hybrid Logic (HL) and argumentation. We argue that model checking provides a significant support in the analysis of a system's model, while the expressivity of HL enables a more refined verification by allowing to focus over specific states or transitions of interest in the model. Once the non-compliant states or transitions are identified, DeLP filters possible repair solutions, considering only the minimum set of changes to be applied to the model such that compliance is achieved.

7.3.1 Illustrative Example

We consider the scenario in [176], referring to the safe insertion of an Unmanned Aircraft Vehicle (UAV) into the civil air traffic. We propose a solution for modeling such Unmanned Aircraft Systems (UASs) in compliance to the set of safety regulations. Let the following set of the "Rules of the Air" dealing with collision avoidance:

R_1: *Obstacle Detection – "All obstacles must be detected within an acceptable distance to allow performing safely the obstacle avoidance maneuver"*

R_2: *Obstacle Avoidance – "All obstacles must be avoided by performing safely a slight deviation from the preestablished path and an immediate return to the initial trajectory once all collision risks are eliminated."*

R_3: *Collision Avoidance – "When two UAVs are approaching each other and there is a danger of collision, each shall change its course by turning to the right."*

The rule R_1 states that all obstacles (i.e. human-controlled aircrafts, other UAVs etc.) that are interfering with the initial trajectory of the UAV must be signaled within a certain limit of time such that to allow avoidance maneuvers to be performed by the UAV in safe conditions. The avoidance maneuver as shown by rules R_2 and R_3 consists of a slight change of the initial path to the right such that to allow the safe avoidance of the approaching UAV followed by a repositioning on the initial trajectory.

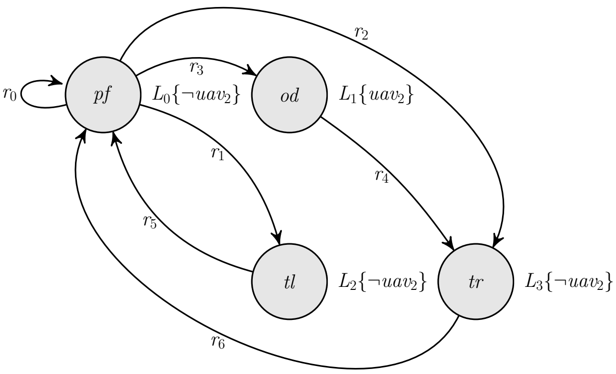


Figure 7.2: Initial Kripke Model $\mathcal{M}_0$ for the UAV.

7.3.2 Kripke Model for the Unmanned Aerial Vehicle

We will further represent the behavior of the UAV noted by uav_1 captured in an obstacle avoidance scenario. The following states will be considered in constructing the Kripke model: path-following (pf), obstacle detection(od), turn left(tl) and turn right(tr). To each state we will attach the boolean state variable uav_2, which will indicate the presence or absence of another approaching UAV. In the path-following state pf, the UAV uav_1 performs a waypoint following maneuver, which includes periodical turns to the left or to the right. The appearance of an obstacle ($uav \rightarrow \top$) leads to the transition of the UAV into obstacle detection state od and from there in turn right tr state as part of the obstacle avoidance maneuver, followed by a return to the initial path-following state. The initial model $\mathcal{M}_0$ is:

$$\mathcal{M}_0 = \langle \{od, tr, tl, pf\}, \{r_0, r_1, r_2, r_3, r_4, r_5, r_6\},$$

$$\{(pf, \{\neg uav_2\}), (od, \{uav_2\}), (tr, \{\neg uav_2\}), (tl, \{\neg uav_2\})\}\rangle$$

Note that $r_0, \ldots, r_6$ are the transitions between states and they belong to R - a family of binary accessibility relations from the Hybrid Kripke structure in Fig. 7.2.

7.3.3 Verifying Compliance to Safety Regulations

Once the modeling of the UAS is done, we have to verify whether the mentioned safety regulations hold for this model. We will further express the two safety regulations using hybrid logics:

The safety regulation R_1 states that once the *od* (*ObstacleDetect*) state is reached then all the successor transitions should contain the transition towards an avoidance maneuver state, for our case here, state *tr*, meaning that the obstacle was detected in time and the avoidance maneuver has been safely performed:

$$R_1 : [F]od \rightarrow tr \tag{7.1}$$

The rule R_2 states that all the following transitions from the *TurnRight* or *TurnLeft* states should always lead to the *PathFollow* state:

$$R_2 : [F](tr \vee tl) \rightarrow pf \tag{7.2}$$

The formula for the safety regulation R_3 states that if another UAV is detected in the *od* (*ObstacleDetect*) state then all the following transitions should be done towards state *tr* (*TurnRight*):

$$R_3 : \bullet_{od}\, uav_2 \rightarrow ([F]od \rightarrow tr) \tag{7.3}$$

7.3.4 Model Repair using Arguments

We focus on the UAV scenario and we illustrate a solution for modeling the existing UAS to comply to newly introduced rules. In this direction, we will consider the initial set of rules extended by a newly adopted norm for UAVs navigating in an Aerodrome Airspace:

R_4: *Navigation in Aerodrome Airspace – "An unmanned aerial vehicle passing through an aerodrome airspace must make all turns to the left [unless told otherwise]."*

First we check whether the existing UAS model complies to the new regulation R_4 translated into Hybrid Logics. To differentiate the contexts for the reasoning process (the presence or the absence of an aerodrome in the vecinity of the UAV), we add to each possible state the boolean variable a, which becomes *true* when the UAV enters an aerodrome airspace:

$$R_4 : @_i\, a \rightarrow ([F]i \rightarrow (\neg tr)) \tag{7.4}$$

The formula translates in natural language as: *all transitions from the states in which the state variable aerodrome a holds should not lead to the tr (TurnRight) state, the only state which is forbidden when navigating inside the aerodrome space.* Since the only states from which turns are possible are *pf* and *od*, we further consider only a reduced subset of states for the verification process. Formula does not hold for the existing model. Considering that the aerodrome a state variable is true for our model, then a turn to the left is not possible from the *od* state, but only to the tr ($TurnRight$) state. Going further, from the pf state transitions are possible to the tl ($TurnLeft$) state, but, at the same time, to the tr ($TurnRight$) states. Therefore, the existing model does not comply to the new regulation. Hence, a separate model $\mathcal{M}_a$ should be considered for UAVs passing through an aerodrome space, which does not include transitions to tr state (see Figure 7.3).

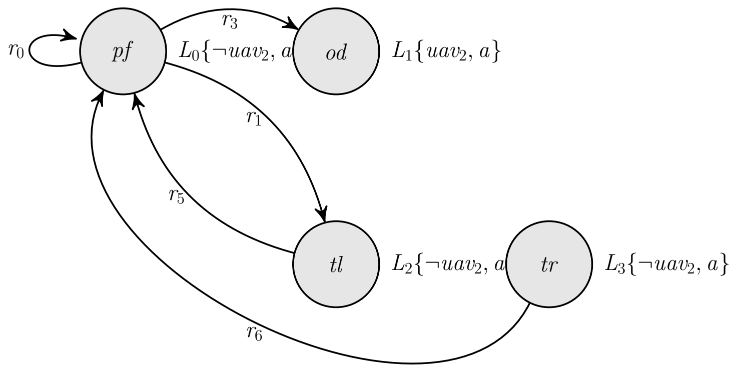


Figure 7.3: Kripke Model $\mathcal{M}_a$ for the UAV in an Aerodrome Space.

In the model in Figure 7.3 there is no possibility to perform a collision avoidance maneuver once an obstacle is detected. Hence, a more refined repairing solution should be applied. In our approach, the initial model $\mathcal{M}_0$ could be extended to include also the new rules without having to construct a new model from the beginning. Although various algorithms were already presented for the repair of a Kripke Model [27], we propose a method based on argumentation for extending the model such that it complies to the updated set of regulations.

To decide upon the most suitable solution (with minimum changes) for model repair, we represent several possible extensions to the Kripke Model as defeasible arguments and include them in DeLP for choosing the best possible option between different conflicting arguments.

First, consider the uav_1 is in the obstacle detect $od \in S$ state, where S is the set of states in $\mathcal{M}$ with the labeling function $L(od) = \{uav_2, \neg a\}$. It means that uav_1 has detected another aerial vehicle uav_2. Assume that in this state the DeLP program will warrant the opposite conclusion a. This triggers the application of the primitive operation PU_3 which updates the labeling function $L(od) = \{uav_2, \neg a\}$ with $L'(od) = \{uav_2, a\}$.

Second, assume that the DeLP program based on the state variables uav_2, and $\neg a$ and the nominal od infers a relation r_i between od and another nominal $i \in \mathcal{N}$

of the model. The repair consists of applying the operation PU_1 on $\mathcal{M}$, where the relation set R' is extended with a relation between the two states ob and i: $R' = R \cup \{(od, i)\}$. The reasoning mechanism is possible because hybrid logic allows to directly refer to the states in the model, by means of nominals.

Third, the program can block the derivation of a relation r between the current state and a next state. For instance, if $L(od) = \{uav_2, a\}$ and the argument A_3 succeeds, the transition between state od and state *turn_right* can be removed. Formally, $R' = R \setminus \{(od, turn\_right)\}$.

Fourth, if the DeLP program warrants, based on the current state variable and available arguments, a nominal i which does not appear in S, the set of states is extended with this state: $S' = S \cup \{i\}$.

7.3.5 Adapting the Model to New Specifications

To apply argumentation for the repair of Kripke models one must be able to map the information encapsulated in the Kripke structure to a DeLP program $\mathcal{P}$ such that arguments can be constructed and based on them updates to be performed on the initial model. In this line, we view the elements of a Kripke structure (states, labels and transition between states) as part of a defeasible logic program $\mathcal{P} = (\Pi, \Delta)$, where the information about states corresponds to the set Π of strict rules, while the labels and the transitions between states belong to the set of defeasible rules Δ. Once a formal verification is performed on the model, which yields a negative result on what it concerns conformance to a certain set of constraints α, we are able to identify whether the presence or absence of a certain state(s) or transition(s) led to the undesired outcome for the model checking task. Depending on the output of the model checker, the following steps are performed:

1. Each non-compliant transition is considered for a query Q_r and an argument $\langle \mathcal{A}_r, Q_r \rangle$ is used to clarify the infringement of a constraint α (promoting or demoting the operation PU_2 to be performed on the model).

2. Each indication of an absence of a required transition leads to a new query Q_{rx} and an argument $\langle \mathcal{A}_{rx}, Q_{rx} \rangle$ which promotes the introduction of the missing transition rx (by performing operation PU_1).

3. Each non-compliant labeling of a state is considered for a query Q_l and an argument $\langle \mathcal{A}_l, Q_l \rangle$ is used to clarify the infringement of the constraint α. It results in an update to the labeling values (by performing operation PU_3).

4. Each indication of an absence of a required state sx leads to an update of the Π set of the defeasible logic program $\mathcal{P}$ by $\Delta^x \sqsubseteq \Delta \cup \{sx\}$ and of the Kripke model (by performing operation PU_4) and the argumentation steps 1-3 are repeated for the updated defeasible program and Kripke model.

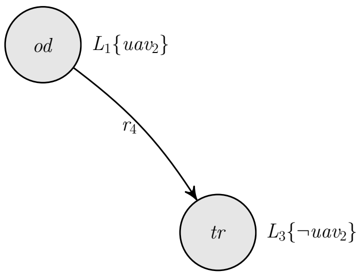


Figure 7.4: Transitions promoted by $\mathcal{A}_2$.

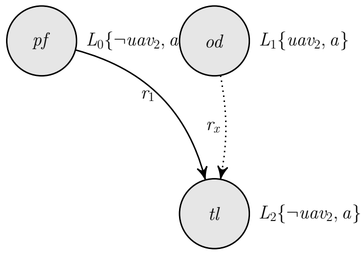


Figure 7.5: Transitions promoted by $\mathcal{A}_3$.

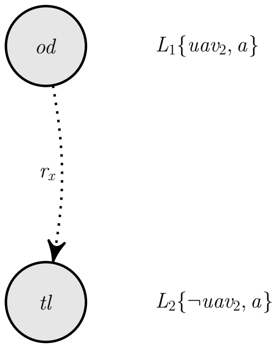


Figure 7.6: Transitions promoted by $\mathcal{A}_4$.

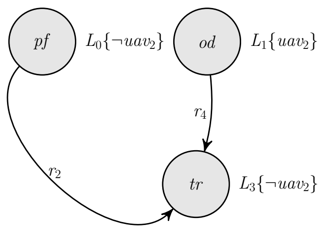


Figure 7.7: Transitions promoted by $\mathcal{A}_5$.

Once the arguments are constructed, the decision over a certain repair solution is taken using DeLP. Going back to our example, we will consider different model updated based on arguments $\mathcal{A}_2$–$\mathcal{A}_5$, promoting the rules R_2, R_3, respectively R_4 depicted in Figs 7.4–7.7.

One can observe by analyzing $\mathcal{M}_0$ that there is no possibility for the UAV to go into the *tl* state once it has reached the *od* state, but only to the *tr* state. Since inside the aerodrome space, only turns to the left are permitted, then the link connecting *od* and *tr* (r_4) should be taken out from the model (see Fig. 7.3). We argue that for compliance to the new regulation, we only need to change the links in the model to point from the *od* and *pf* states only to the *tl* state, when the state variable a is set to true (indicating the presence of an aerodrome in the vicinity of the UAV). Hence, we need to perform the following PU operations for updating the model:

1. (PU_2) Remove the relations (od, tr) and (pf, tr) such that we have: $S' = S$, $L' = L$, and $R' = R \setminus \{(od, tr), (pf, tr)\}$ as indicated by $\mathcal{A}_3$ (in Fig 7.5).

2. (PU_1) Add the relation (od, tl) such that we have: $S'' = S'$, $L'' = L'$, and $R'' = R' \cup \{(od, tl)\}$ as indicated by $\mathcal{A}_3$ and $\mathcal{A}_4$ (see Fig 7.5 and 7.6).

However, the remove operation should be necessary only when that specific relation element causes a conflict between two arguments.

We further consider a new argument $\langle \mathcal{A}_6, \mathit{alter\_course}(\mathit{uav_1}, \mathit{left})\rangle$, which suggests updating rule R_3 by allowing the obstacles to be avoided to the left, instead of to the right when inside the aerodrome space, where:

$$\mathcal{A}_6 \;=\; \left\{ \begin{array}{l} \mathit{alter\_course}(\mathit{uav_1}, \mathit{left}) \prec \mathit{aircraft}(\mathit{uav_1}), \mathit{aircraft}(\mathit{uav_2}) \\ \qquad \mathit{collision\_hazard}(\mathit{uav_1}, \mathit{uav_2})\mathit{nearby}(\mathit{uav_1}, \mathit{aerodrom}) \\ \mathit{collision\_hazard}(\mathit{uav_1}, \mathit{uav_2}) \prec \mathit{approaching\_head\_on}(\mathit{uav_1}, \mathit{uav_2}), \\ \qquad \mathit{distance}(\mathit{uav_1}, \mathit{uav_2}, X), X < \mathit{1000} \end{array} \right\}.$$

If we go back to argument $\mathcal{A}_2$, promoting the application of the initial rule R_2 and $\mathcal{A}_6$, sustaining a slight modification of the rule R_2 for navigation in aerodrome space, one can see that they do not attack each other as they offer solutions for different contexts: the $\mathcal{A}_2$ argument refers to collision avoidance outside the aerodrome space, while the $\mathcal{A}_6$ argument considers the case of collision avoidance when the UAV is nearby an aerodrome. A similar reasoning applies for the transition (pf, tr), which will be possible only when the state variable a does not hold at pf. Therefore, the PU_2 step can be left out and the updating of the model can be done only through a PU_1 operation. The decision to turn left or turn right will be taken in accordance to the value of the state variable a, which indicates the presence or absence of an aerodrome in the vicinity of the UAV.

We illustrate the update operation by adding a link r_7 from the od to the tl state. Additionally, we attach to each state the boolean state variable a, such that it allows the UAV to perform only those transitions that comply to the set of regulations in different contexts, respectively inside or outside the aerodrome space. The updated model $\mathcal{M}_x$ is presented in Fig. 7.8.

$$\mathcal{M}_1 = \langle \{od, tr, tl, pf\}, \{r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7\},$$

$$\{(pf, \{\neg uav_2\}), (od, \{uav_2\}), (tr, \{\neg uav_2, \neg a\}), (tl, \{\neg uav_2\})\}\rangle$$

If the UAV reaches the od state, then it performs the transition to the next state that has the same value for a as the od state. Thus, if uav_1 detects another approaching uav_2 and it is outside the aerodrome space ($\neg a$), it looks for the next possible state that has the same value for the variable a. Fig. 7.8 shows that the state that complies to this condition is tr. Also, if uav_1 is in the pf state and the variable a holds at that state, then the possible transitions will be tl or od.

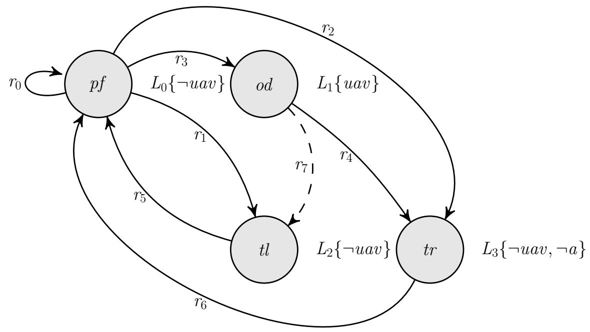


Figure 7.8: Extended model $\mathcal{M}_x$ for the UAV compliant with the new regulation.

If uav_1 reaches the *od* state, while in the vicinity of an aerodrome, it performs a transition to the *tl* state, where the state variable a also holds. If uav_1 reaches *pf* then it will perform a transition to either *tl* or *od* states. The other transitions from the model are not dependent on the state variable a, therefore they will remain the same as in the initial model. By adding the condition $\neg a$ for reaching state *tr*, we can avoid transitions to that state when a holds for the model.

7.4 Conclusions

We presented a framework based on defeasible argumentation and model checking that is suitable to develop of recommender systems for assisting flight controllers in air traffic control systems. Our approach presents a serious aid for assisting flight controllers to reach decisions related to safety constraints. We presented a case study where Defeasible Logic Programming is used to codify a set of possibly incomplete and potentially inconsistent landing safety criteria. The data is evaluated in a real-time fashion to provide a stream of safety recommendations based on the input fed to the system by the plane and runway sensors located in a simplified world of an airport with a plane that have to land under different conditions. We showed that model checking provides a significant support in the analysis of a system's model, while the expressivity of Hybrid Logics used in formalizing different constraints that the system must comply to, enables a more refined verification. by allowing to focus over specific states or transitions of interest in the model. Once the non-compliant states or transitions are identified, DeLP provides a filtering between possible repair solutions, considering only the minimum set of changes to be applied to the model such that compliance is achieved. To apply argumentation for the repair of Kripke models, we map the information encapsulated in the Kripke structure to a DeLP program. The elements of a Kripke structure (states, labels and transition between states) are considered part of a defeasible logic program, where the information about states corresponds to the set of strict rules, and the labels and the transitions between states can be regarded as belonging to the set of defeasible rules. Once a formal verification is performed on the model, which yields a negative result on what it concerns conformance to a certain set of constraints, we identify whether the presence or absence of a certain state or transition led to the undesired outcome for the model checking task. Depending on the output of the model checker, the following steps are performed: each non-compliant transition is considered for a query and an argument is entailed for clarifying the infringement of a constraint; each indication of an absence of a required transition leads to a new query and an argument which promotes the introduction of the missing transition; each non-compliant labeling of a state is considered for a query and an argument is entailed for clarifying the infringement of the constraint, which results in an update to the labeling values, and, each indication of an absence of a required state leads to an update of the set of the defeasible logic program by and of the Kripke model and the argumentation steps are repeated for the updated defeasible program and Kripke model. Once the arguments are constructed, the decision over a certain repair solution is taken using DeLP.

The presented approach has several contributions in regards to the field of Expert and Intelligent Systems. To the best of our knowledge, this is the first approach that combines defeasible argumentation and model checking to the problem of dealing with inconsistent and incomplete safety criteria. Because our approach is based on Defeasible Logic Programming, the inconsistency of criteria is handled automatically by the reasoning system. Therefore, the system engineer can concentrate on the knowledge representation process even when the field being modeled could be intrinsically inconsistent, and when the field is consistent, the system would behave exactly the same as a traditional logic programming setting. In our particular case study, the arguments produced by the argumentation reasoning mechanism are compared syntactically using specificity, however, the system accepts other modular criteria for comparing arguments, making it flexible for modeling other comparison criteria (e.g. based on measures of sensor reliance, trust between different sources, etc.).

For implementing continuous reasoning in DeLP, the embedding of infinite list processing in the world of logic programming can benefit with a functional programming approach based on lazy evaluation as suggested by [66].

Chapter 8

Normative deontic logic

To enable compliance checking on integrated business processes we developed the $\mathcal{NTL} - \mathcal{ALC}$ logical framework, for closing the gap between the abstract norms and the concrete business processes. To reason on the active obligations and permissions, we extended the Normative Temporal Logic ($\mathcal{NTL}$), by applying the deontic operators O (obligation) and P (permission) on concepts of the $\mathcal{ALC}$ (Attribute Language with Complements) description logic. As proof of concept of our results we have used the Hazard Analysis at Critical Control Points (HACCP) standard, aiming to prevent the occurrence of significant hazards in the food industry.

8.1 Motivation

We address the problem in the representational gap between normative specifications and process models [54]. While regulatory systems consist of high level, sometimes principle-based specifications acting more like a framework, business processes are defined in terms of low level, specific atomic tasks. Our solution extends the semantics of the normative temporal logic [2], applying it over $\mathcal{ALC}$, a Description Logic (DL) language, instead of propositional variables. Even if the integration of semantic knowledge in the process management technology is considered a milestone in compliance checking [121], semantic modeling of norm compliance is unfortunately still highly ignored by current approaches [94]. As a proof of concept scenario, we have considered a shrimp processing workflow governed by the HACCP standard, where additional knowledge about shrimps, possible hazards, or health consequences is available. For testing the above concepts a Lisp-based prototype was built on top of the ACL2 theorem prover, and for the DL reasoning services the Racer system was integrated.

Rule compliance is important in the path towards recovery from the current economic crisis [29, 137, 30]. The need of compliance related software has gained at-

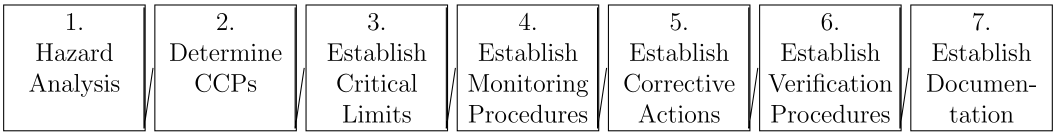


Figure 8.1: HACCP principles

tention [119, 34] as a feasible solution to one of the causes of the crisis: norm breach by human agents in critical domains, aiming to avoid a set of financial scandals such as Enron, Worldcom, Parmalat [135]. Compliance management faces different types of regulations: i) *regulatory approach*, covering mandatory legal regulations (SOX, EUROSOX, MIFID) as well as corporate governance (Sodd Responsibility) and ii) *standardization approach*, including certifications (ISO, HACCP) or corporate standard best practices (ITIL, Cobit) [92]. In this line, certification has become a competitive advantage among business entities.

8.2 Requirements for HACCP-based systems

HACCP [44] is based on the seven steps (Fig. 8.1). During *hazard analysis* business entities determine the food safety hazards and identify the preventive measures for controlling them. In the second step, *critical control points* (CCPs) are identified. A CCP is a step-point in a food process at which a specific action can be applied in order to prevent a safety hazard, to eliminate it, or to reduce it to an acceptable level. The procedure continues at step 3 by *establishing critical limits*. A critical limit is a criterion which separates acceptability from unacceptability. The criteria often used include measurements of time, temperature, moisture level, pH, Aw, available chlorine, and sensory parameters such as visual appearance. Critical limits must be specified for each CCP. Monitoring is the scheduled observation of a CCP related to its critical limits. At step 4 the monitoring procedures must be able to detect the loss of control at the CCP. The corrective actions at step 5, to be taken when monitoring indicates that a particular CCP is not under control, are specified for each CCP to deal with the deviations observed. At the 6th step, verification and auditing methods, procedures and tests, including random sampling and analysis, can be used to determine if the HACCP system is working correctly. *Establishing documentation* at step 7 concerns all procedures and records appropriate to the application of these principles.

8.2.1 Identifying requirements for HACCP-based systems

Domain knowledge. A clear mapping should be provided between the general regulations and the running business processes. Lack of expertise was identified as the main barrier for the effective implementation of HACCP in small and medium enterprises [13, 168]. The HACCP involves reasoning within several domains: medical (health status, possible illnesses, possible causes, possible symptoms, allergies); legal (current regulations, safety norms, legal consequences); specific knowledge about the subject of activity (food industry); and engineering.

Critical limits violation. The theory of "contrary to duty obligations" maps perfectly over the fourth principle of the HACCP: one has the obligation to keep the items within the pre-defined critical limits, but when this is breached, the obligation to apply specific control measures holds. For instance, in case of inadequate cooking time, the obliged control measure is either recook or hold and evaluate.

The main conclusion is that even if the spread of the HACCP systems in industry is ubiquitous[1], the existing support tools are rudimentary and they do not enhance or augment the reasoning capabilities of an actor by providing both background knowledge and compliance checking. One reason is the soup of technologies that need to be merged in order to manage a complex structured HACCP implementation, which our approach tries to overcome.

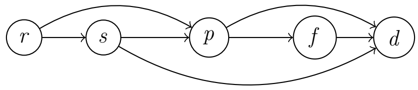
8.3 Normative temporal logic over $\mathcal{ALC}$

To reason on the active obligations and permissions in an HACCP system, we extend the logic introduced in [2] by applying the deontic operators O and P on concepts of the $\mathcal{ALC}$ (Attribute Language with Complements) description logic instead of propositional variables of classical logic. The normative temporal logic ($\mathcal{NTL}$) [2] is a modified version of the branching time temporal logic CTL, in which the path quantifier A ("on all paths") is replaced with the indexed deontic operator O (obligation) and E ("on some path") is replaced by the deontic operator P (permission). The resulted $\mathcal{ALC} - \mathcal{NTL}$ is defined over Kripke Structures.

The semantics of the $\mathcal{NTL} - \mathcal{ALC}$ introduced here is based on sequences of $\mathcal{ALC}$ interpretations over the state dependent domain $\Delta^{I(s)}$ (constant domain assumption does not hold). The interpretations satisfy the unique name assumption within all domains $\Delta^{I(s)}$, where s represents a state in the Kripke structure.

Definition 35 *An interpretation for the state s is a pair $I(s) = (\Delta^{I(s)}, \cdot^{I(s)})$ where the domain $\Delta^{I(s)}$ is a non-empty set, and $\cdot^{I(s)}$ is a function that assigns to every concept name A a set $A^{I(s)} \subseteq \Delta^{I(s)}$, to every role r a binary relation $r^{I(s)} \subseteq \Delta^{I(s)} \times \Delta^{I(s)}$, and to every individual a an element $a^{I(s)} \in \Delta^{I(s)}$.*

[1]The European Union issued a regulation in January 2006, requesting all food operators to implement a food safety system based on HACCP principles.



$\Delta^{I(r)} = \{metabisulfite, reject\}$ $PH^{I(r)} = \{metabisulfite\},\ CM^{I(r)} = \{reject\}$ $hasControlMeasure^{I(r)} = \{(metabisulfite, reject)\}$
$\Delta^{I(s)} = \{salmonella, shrimps\_lot1, -5\,^\circ C\}$ $PH^{I(s)} = \{salmonella\}$ $hasTemperature^{I(s)} = \{(shrimps\_lot1, -5\,^\circ C\}$
$\Delta^{I(p)} = \{salmonella\_growth, high\_temperature\_cooking, 70\,^\circ C\}$ $PH^{I(p)} = \{salmonella\_growth\},\ CM^{I(p)} = \{high\_temperature\_cooking\}$ $hasControlMeasure^{I(p)} = \{(salmonella\_growth, high\_temperature\_cooking)\}$ $hasTemperature^{I(p)} = \{(shrimps\_lot1, 70\,^\circ C)\}$
$\Delta^{I(f)} = \{shrimps\_lot1, -18\,^\circ C\}$ $hasTemperature^{I(f)} = \{(shrimps\_lot1, -18\,^\circ C)\}$
$\Delta^{I(d)} = \{shrimps\_lot1, monday\}$ $Time^{I(d)} = \{monday\}$ $shipment^{I(d)} = \{(shrimps\_lot1, monday)\}$

Figure 8.2: Kripke structure for shrimp processing. *PH*, *CM*, and *Time* represent concepts. *hasControlMeasure*, *hasTemperature*, and *shipment* represent roles.

Definition 36 (Semantic Kripke structure) *A semantic Kripke structure over domain* Δ *is a quad* $\mathcal{K} = \langle S, S_0, R, I\rangle$*, where:* S *is a finite, non-empty set of states, with* $S_0 \subseteq S$ $(S_0 \neq \varnothing)$ *as initial states;* $R \subseteq S \times S$ *is a total binary transition relation on* S*; and the function* I *associates with each state* $s \in S$ *an* $\mathcal{ALC}$ *interpretation* $I(s) = (\Delta^{I(s)}, \cdot^{I(s)})$ *where* $\bigcup \Delta^{I(s)} = \Delta$*, such that* $\forall\, s, s' \in S$*: (i)* $\forall\, r \in \Delta \times \Delta, r^{I(s)} = r^{I(s')}$ *(rigid roles); and (ii)* $\forall\, a \in \Delta, a^{I(s)} = a^{I(s')}$ *(rigid individual names).*

Given the TBox Θ, each local interpretation $I(s)$ associates an atomic concept $A \in \Theta$ with a set of domain objects $A^{I(s)} \in \Delta^{I(s)}$ and each atomic role $r \in \Theta$ with a set of pairs $r^{I(s)} \in \Delta^{I(s)} \times \Delta^{I(s')}$. Informally, the function $I(s)$ associates each state with the set of concepts from Θ which are satisfied in that state based on the individuals enumerated in $\Delta^{I(s)}$, such that once a role or an individual is designated in a state, it will not change its name over time. Note that the rigid concept assumption $\forall\, A \in \Theta, A^{I(s)} = A^{I(s')}$ does not hold. The rationality behind this is illustrated by example 31.

Example 31 *The shrimps are received (r), ice stored (s), processed (p), frozen (f), and delivered (d) to the retailer (see the graph in figure 8.2). The business process states that the received shrimps can be processed directly (r* $\rightarrow$ *p)* *or ice stored at $-5\,^{\circ}C$ for future processing ($r \rightarrow s$), and they can be delivered to the retailer as frozen raw shrimps at $-20\,^{\circ}C$ ($s \rightarrow d$) or already processed ($p \rightarrow d$). The processed shrimp may be frozen if the delivery does not occur immediately ($p \rightarrow f$). These frozen shrimps will eventually be delivered ($f \rightarrow d$). The corresponding Kripke structure $\mathcal{K} = \langle \{r, s, p, f, d\}, \{r\}, R = \{(r, s), (r, p), (s, p), (s, d), (p, d), (p, f), (f, d)\}, I\rangle$ has the interpretation function I extensionally listed in figure 8.2. Here PH stands for PotentialHazard and CM for ControlMeasure. The available data in each state represents the domain of interpretation for that particular state. For instance $\Delta^{I(r)}$ has two individuals, metabisulfite and reject, where the first one is an instance of the PH concept, and the second one an instance of the CM concept.*

Two observations follow: Firstly, note that the concepts *PH* and *CM* are not rigid ($PH^{I(r)} \neq PH^{I(s)} \neq PH^{I(p)}$). Secondly, the control measure for salmonella identification is the individual *high_temperature_cooking* which occurs only in the state p. This is a step towards a *constructive interpretation* [129] of description logic. Based on the rigid individual names assumption, by unfolding the $\mathcal{K}$ structure to the corresponding tree, it follows that an individual labeled in a state s carries forward its label in the unfolding of each successor state s', given by $\Delta^{I(s)} \subseteq \Delta^{I(s')}$, where s is the ancestor of s'.

The syntax of $\mathcal{NTL} - \mathcal{ALC}$ is defined by the following grammar.

$$C, D ::= \top \mid A \mid \neg C \mid C \sqcup D \mid \exists\, r.C \mid \forall\, r.C \mid P(\odot C) \mid P(C \mathcal{U} D) \mid O(\odot C) \mid O(C \mathcal{U} D)$$

The semantics is given with respect to the satisfaction relation "$\models$" and the interpretation function I. Let Θ be a TBox describing a specific business activity. A concept $C \in \Theta$ is satisfied in the state s if there is an interpretation $I(s)$ so that $C^{I(s)} \neq \varnothing$. We note by $\pi[u]$ the state indexed by u in the path π. $\Gamma(s)$ is the set of all paths starting from the state s. $\mathcal{K}, s \models_I A$ holds when $\mathcal{K}$ is a semantic Kripke structure, s a state in $\mathcal{K}$, I an interpretation over $\mathcal{K}$ and A an atomic concept in Θ, as follows.

$\mathcal{K}, s \models_I \top$
$\mathcal{K}, s \models_I A$ iff $A^{I(s)} \neq \varnothing$ or $\exists\, B^{I(s)} \neq \varnothing$ where $B \sqsubseteq A \in \Theta$
$\mathcal{K}, s \models_I \neg C$ iff $\mathcal{K}, s \not\models_I C$
$\mathcal{K}, s \models_I \exists\, r.C$ iff $\{a \in \Delta^{I(s)} \mid \exists\, b \in \Delta^{I(s')} : (a, b) \in r^{I(s)} \wedge b \in C^{I(s')}\} \neq \varnothing$
$\mathcal{K}, s \models_I C \sqcup D$ iff $\mathcal{K}, s \models_I C$ or $\mathcal{K}, s \models_I D$
$\mathcal{K}, s \models_I O(\odot C)$ iff $\forall\, \pi \in \Gamma(s) : \mathcal{K}, \pi[1] \models_I C$
$\mathcal{K}, s \models_I P(\odot C)$ iff $\exists\, \pi \in \Gamma(s) : \mathcal{K}, \pi[1] \models_I C$
$\mathcal{K}, s \models_I O(C \mathcal{U} D)$ iff $\forall\, \pi \in \Gamma(s)\ \exists\, u \in \mathbb{N}\ s.t.$
$\mathcal{K}, \pi[1] \models_I D \wedge \forall\, v \in [0, u) \mathcal{K}, \pi[v] \models_I C$
$\mathcal{K}, s \models_I P(C \mathcal{U} D)$ iff $\exists\, \pi \in \Gamma(s)\ \exists\, u \in \mathbb{N}\ s.t.$
$\mathcal{K}, \pi[u] \models_I D \wedge \forall\, v \in [0, u) \mathcal{K}, \pi[v] \models_I C$

Ax0	all validities of ALC logic
Ax1	$P(\odot C) \Rightarrow P(\odot D)$, where $D \sqsubseteq C \in \Theta$
Ax2	$O(\odot\neg C) \Rightarrow O(\odot\neg D)$, where $D \sqsubseteq C \in \Theta$
Ax3	$P \odot (C \sqcup D) \Leftrightarrow P(\odot C) \vee P(\odot D)$
Ax4	$P(\Diamond A) \Rightarrow P(\top \mathcal{U} B)$, where $B \sqsubseteq A \in \Theta$
Ax5	$O(\Box A) \Rightarrow \neg P(\Diamond\neg B)$, where $B \sqsubseteq A \in \Theta$
Ax6	$P(\Box A) \Leftrightarrow \neg O(\Diamond\neg B)$, where $B \sqsubseteq A \in \Theta$
Ax7	$P(\odot A) \vee P(\odot B) \Rightarrow P(\odot E)$, where $E \sqsubseteq A \sqcup B \in \Theta$
Ax8	$O(\odot A) \rightarrow \neg P(\odot\neg B)$, where $B \sqsubseteq A \in \Theta$
PrezA	$O(\odot C \vee \Diamond D) \wedge \odot\neg E \wedge \neg O(\odot C) \Rightarrow O(\Diamond D)$, where $C \sqsubseteq E$

Figure 8.3: Axioms of $\mathcal{NTL} - \mathcal{ALC}$.

The semantics of the "sometime in the future operator" $\Diamond$ and in all future states along a path $\Box$ are introduced as usual.

$$O\Diamond C \equiv O(\top \mathcal{U} C) \qquad P(\Diamond C) \equiv P(\top \mathcal{U} C)$$
$$O\Box C \equiv \neg P(\odot\neg C) \qquad P(\Box C) \equiv \neg O \odot \neg C)$$

The semantics is driven by the interpretation function I instead of the labeling function in the classical Kripke structures. Informally, the concept A is satisfied in the state s if there exists a sub-concept B according to Θ, so that the interpretation of B is not the empty set in state s. The proposed semantics for $O(\odot C)$ stipulates that in the next state there is the obligation that the concept C should be satisfied by the business process modeled by $\mathcal{K}$. The obligation is satisfied iff in the next state an individual which satisfies the interpretation I of the concept C (or of a sub-concept $D \sqsubseteq C$) exists. The inference system is given by the following rules:

$$r_1 : \frac{\mathcal{K}, s \models_I A, A \sqsubseteq B}{\mathcal{K}, s \models_I B}, r_2 : \frac{\mathcal{K}, s \models_I (A \Rightarrow O\Diamond L), B \sqsubseteq A, \mathcal{K}, s \models_I B}{\mathcal{K}, s \models_I O\Diamond L}$$

where $\odot$ stands for a temporal operator in $\{\odot, \Diamond, \Box\}$.

The normative reasoning based on the axioms shown in the figure 8.3 is illustrated in table 8.1. As an example, in cooking and the ways to cook shrimps, axiom 1 says that if you have permission to cook, then it implies that you may use any method of cooking (such as Stream Cooking) as long as that method is found in the domain knowledge Θ. Similarly for axiom 2, if we are talking about the preservative concept, then when the obligation is not to introduce preservatives, the business entity is also aware not to use sulfites, because they are a kind of preservative as specified in the domain ontology. The corresponding version of Ax2 for positive concepts, given by $O(\odot C) \Rightarrow O(\odot D)\ D \sqsubseteq C$, does not hold. As a motivation, assume the obligation to cook the shrimps holds for the next state,

Domain knowledge Θ	**Normative reasoning**	**Based on**
$SteamCook \sqsubseteq Cook$	$P(\odot Cook) \Rightarrow P(\odot SteamCook)$	Ax1
$Sulfites \sqsubseteq Preservatives$	$O(\odot\neg Preservatives) \Rightarrow O(\odot\neg Sulfites)$	Ax2
	$P(\odot(SteamCook \sqcup Refrigerate)) \Leftrightarrow$ $P(\odot SteamCook) \vee P(\odot Refrigerate)$	Ax3
$FrozenStorage \sqsubseteq ColdStorage$	$P(\Diamond ColdStorage) \Rightarrow P(\top \mathcal{U} FrozenStorage)$	Ax4
$Clean \sqsubseteq \neg Impurity \sqcap \neg Dirt$	$O(\Box Clean) \Rightarrow \neg P(\Diamond Impurity)$	Ax5
$SteamCook \sqsubseteq Cook$	$P(\Box Cook) \Leftrightarrow \neg O(\Diamond\neg SteamCooked)$	Ax6
$E \sqsubseteq A \sqcup B$	$P(\odot A) \vee P(\odot B) \Rightarrow P(\odot E)$	Ax7
$PreservativeLabel \sqsubseteq Label$	$O(\odot Label) \Leftrightarrow \neg P(\odot\neg PreservativeLabel)$	Ax8

Table 8.1: Filling the gap between general norms and the specific business process.

given by $O(\odot Cook)$, and that the food operator knows that shrimps can be boiled, steamed, or grilled ($BoilCook \sqsubset Cook$, $SteamCook \sqsubset Cook$, $GrillCook \sqsubset Cook$). The positive version of Ax2 would force the agent to apply all the subsumed cooking methods in the next state. A different form of negation is needed to extend the axioms with negated forms.

The value of the axioms is that they allow business agents to infer their specific permissions and obligations from abstract normative specifications, since the right hand side of each axiom refers to a more specific concept in the knowledge base Θ. By chaining the axioms, one is able to minimize the gap between the high level, general, specification of the normative system and the low level, specific business process: $O(\odot\neg Preservatives) \Rightarrow O(\odot\neg Sulfites) \Rightarrow O(\odot\neg Metabisulfite)$. The effect of the PrezA axiom in figure 8.1 is illustrated in example 32.

Example 32 *Suppose that at a step in the processing flow the food operator should either steam cook the shrimps in the next state or to refrigerate the item in some future step for future processing:* $O(\odot SteamCook \sqcup \Diamond Refrigerate)$. *Assuming that in the next state the items were not steam cooked* ($\odot\neg SteamCooked$) *and that there is no obligation to cook the shrimps* ($\neg O(\odot Cook)$)*, under the subsumption* $SteamCook \sqsubseteq Cook$*, the PrezA axiom in table 8.1 states the obligation to refrigerate the item in some future step, given by* $O(\odot SteamCook \vee \Diamond Refrigerate) \wedge \odot\neg SteamCooked \wedge \neg O(\odot Cook) \Rightarrow O(\Diamond Refrigerate)$.

8.4 Normative systems

In a supply chain scenario the structure $\mathcal{K}$ encapsulates the processing steps or possible workflow that an item follows until it becomes a finite product, while the active norms represent the additional constraints or standards of quality imposed by different members of the chain.

Definition 37 *Given a business process $\mathcal{K}$ and the* TBox Θ*, a norm $\eta\langle C, D\rangle$, $C, D \in \Theta$, is the set of pair of states (s, s') where $C^{I(s)} \neq \varnothing$ and $D^{I(s')} \neq \varnothing$.*

The meaning of the pair $(s, s') \in \eta\langle C, D\rangle$ is that the transition (s, s') is forbidden in the context of $\eta\langle C, D\rangle$. Thus, if the concept C is satisfiable in state s and D is satisfied in s' the transition (s, s') is not acceptable in the business process $\mathcal{K}$, given the semantic knowledge Θ.

Example 33 *Consider the norm $\eta_1\langle NotFrozen, Delivery\rangle$: "before the delivery all the items should be frozen". Assume the domain knowledge $\Theta = \{Delivery \sqsubseteq \exists\, shipment.Time, NotFrozen \sqsubseteq \exists\, hasTemperature.AboveZero\}$ is active for the business process in figure 8.2. Delivery holds only for state d, given by $Delivery^{I(d)} = \{shrimp\_lot1\}$, whilst for all states $\sigma \neq d$, $Delivery^{I(\sigma)} = \varnothing$. NotFrozen holds for state p since shrimp_lot1 $hasTemperature = 70\,^{\circ}C$. The other two states s and f having Delivery as the next state, all show temperature less than zero: $NotFrozen^{I(s)} = \varnothing$ and $NotFrozen^{I(f)} = \varnothing$. Consequently, the only removed transition by the norm η_1 is (p, d). Formally $\eta_1\langle NotFrozen, Delivery\rangle = \{(p, d)\}$.*

Usually, business entities follow simultaneously different normative systems, aiming to regulate activities at specific processing steps or to guarantee a specific standard of quality to each partner. The normative systems are specified based on the level of expertise and available knowledge, which may not be uniformly distributed among the agents in the supply chain. Usually operators have specific knowledge related to a domain and more general or fuzzy knowledge about other domains. For instance, the actors at the end of the chain (retailers, shipment agents) act upon normative systems encapsulating refined obligations and permissions regarding consumers and related services offered for them, but they might have only general principles or abstract norms related to ingredients of the products they sell. Quite the opposite, the agents at the beginning of the chain follow specific standards for processing food technologies or handling safety hazards, and limited or indirect norms with respect to the end user requirements or complaints.

Definition 38 *A normative system Ω is a collection of norms $\{\eta_1, \eta_2, ...\eta_n\}$ such that $R \setminus I(\eta_i)$ is a total relation simultaneous for all norms η_i, where $I : \Delta \rightarrow 2^R$ is the interpretation function of the norm η_i.*

If $\Omega \subset \Omega'$, then Ω places fewer constraints on a system than Ω', hence the normative system Ω is more liberal [2]. Consider that a product should be simultaneously conformant with two norms η_1 and η_2, which is equivalent to saying that the item should obey the more restrictive normative system $\Omega' = \{\eta_1, \eta_2\}$. When the item should obey the common normative constraints imposed by η_1 and η_2, we say that the normative system $\Omega'' = \{\eta_1 \cap \eta_2\}$ is more liberal. The following axioms are

Algorithm 4: HACCP compliance checking.

```
Input: Θ′(T_HACCP, A_HACCP), Θ″(T_BD, A_BD), Θ‴(T_BW, A_BW), Σ, f:ntl-alc
Output: {s:ProcessingStep) | Θ‴, s ⊭ f }
S = ∅, S_0 = ∅, R = ∅
foreach i ∈ A_BW do
    if i:ProcessingStep then S := S ∪ {i}; labels(i) = ∅; Δ^{I(i)} = ∅;
    foreach s ∈ S do
        if ¬ s:ProcessingStep ⊓ ∃ hasPastStep.ProcessingStep then S_0 := S_0 ∪ {s};
        foreach s′ ∈ S do
            if related(s,s', hasNextStep) then R := R ∪ {(s, s′)};
        end
    end
end
foreach i ∈ Δ^{I(s)} do
    foreach C ∈ T_HACCP ∪ T_BD do
        if i:C then labels(s) := labels(s) ∪ {C};
    end
end
foreach η⟨A, B⟩ ∈ Σ do
    foreach (s, s′) ∈ R do
        if A ∈ labels(s) ∧ B ∈ labels(s′) then R := R \ {(s, s′)};
    end
end
if ntl-alc(Θ‴, s, f) then s ∈ BreachedStates;
```

added to the $\mathcal{NTL} - \mathcal{ALC}$ to express these ideas: Ax9: $\Omega \sqsubseteq \Omega' \Rightarrow (O_\Omega A \Rightarrow O_{\Omega'} A)$, Ax10: $\Omega \sqsubseteq \Omega' \Rightarrow (P_{\Omega'} A \Rightarrow P_\Omega A)$.

Definition 39 *An integrated normative system of a business process is a tuple* $\langle \Theta, \mathcal{K}, \Sigma \rangle$, *where* Θ *is a knowledge base formalizing the business process* $\mathcal{K}$ *and* Σ *is the set of normative systems active for the same business process.*

The semantics of the compliance checking (figure 8.4) covers four main steps: i) introducing the domain knowledge; ii) generating the Kripke semantic structure $\mathcal{K}$ for the running business process; iii) applying the active normative systems Ω_i on the current workflow R; iv) checking the compliance based on conformance patterns formalized in $\mathcal{NTL} - \mathcal{ALC}$.

The algorithm 4 details the steps above. The input consists of the ontology Θ' representing the knowledge related to the HACCP standard, the ontology Θ'' formalizing the business domain and the ontology Θ''' describing the current business workflow. Usually, the normative system Σ represents the internal quality control,

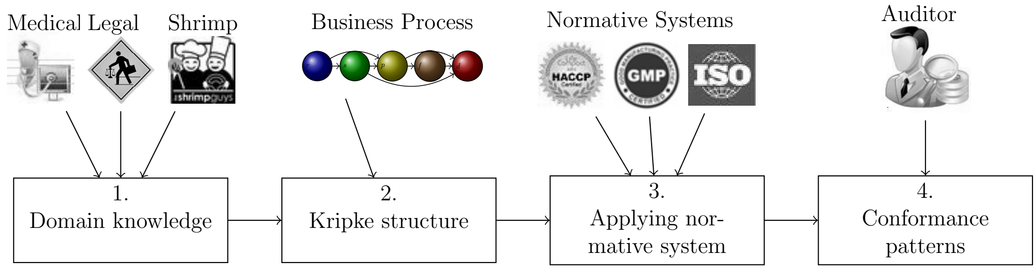


Figure 8.4: Steps for semantic compliance checking.

while the f formalized in the $\mathcal{NTL} - \mathcal{ALC}$ logic is given by the external auditor. The output represents the set of business steps where the formula f is breached.

In the first step, the Kripke structure is generated from the workflow description. Thus, for each individual $i \in A_{BW}$ of type processing step, a state in the Kripke structure is generated. The set S_0 of the initial states contains those individuals which are not related with any other individual by the role *hasPastStep*. Its inverse role, *hasNextStep*, is used to generate the transition relation R. For labeling each state, the algorithm considers all individuals in the current domain of interpretation $\Delta^{I(s)}$. For each such individual i, the state s is labeled with all the concepts C in the quality standard T_{HACCP} or business domain T_{BD} for which i is an instance. Next, the internal control eliminates the transition (s, s') where for each norm $\eta\langle A, B\rangle$ in the current normative system, the concept A is satisfied in s, while B is satisfied in s'. Finally, the function $ntl - alc$, given the Kripke structure corresponding to the Θ''' business workflow, checks that formula f holds in the state s. If not, the state s is added to the breached states.

8.5 Integration of the normative systems

This section follows the above procedure to check the compliance of a realistic shrimp processing workflow against the HACCP normative system.

8.5.1 Ontologies for HACCP views

At the top level HACCP ontology Θ', hazards are defined as biological, chemical, or physical agents that are likely to cause illnesses or injuries to the consumer (line 1 in figure 8.5). The definition is refined in line 2, where biological hazards include harmful bacteria, viruses, or parasites. Chemical agents include compounds that can cause illnesses or injuries due to immediate or long-term exposure. Physical hazards (line 4) are either foreign materials unintentionally introduced in food products (ex: metal fragments in minced meat) or naturally occurring objects

1. $\mathit{Hazard} \equiv (\mathit{Biological} \sqcup \mathit{Chemical} \sqcup \mathit{Physical}) \sqcap$
   $(\exists\, \mathit{causeIllness}.\mathit{Consumer} \sqcup \exists\, \mathit{causeInjury}.\mathit{Consumer})$
2. $\mathit{Biological} \equiv \mathit{Bacteria} \sqcup \mathit{Viruses} \sqcup \mathit{Parasites}$
3. $\mathit{BacterialContamination} \sqsubseteq \mathit{Bacteria},\ \mathit{BacterialGrowth} \sqsubseteq \mathit{Bacteria}$
3. $\mathit{PathogenContamination} \sqsubseteq \mathit{Bacteria},\ \mathit{PathogenGrowth} \sqsubseteq \mathit{Bacteria}$
4. $\mathit{Physical} \equiv \mathit{ForeignMaterial} \sqcup (\mathit{NaturrallyOccuringObject} \sqcap \exists\, \mathit{hasThreat}.\mathit{Consumer})$
5. $\mathit{fishBones} : \mathit{NaturralyOccuringObject}$
6. $\mathit{ForeignMaterial} \equiv \mathit{Glass} \sqcup \mathit{Metal} \sqcup \mathit{Plastic} \sqcup \mathit{Stone} \sqcup \mathit{Wood}$
7. $\mathit{glassBulb} : \mathit{Glass}$
8. $\mathit{ControlMeasure} \sqsubseteq \exists\, \mathit{hasCriticalLimit}.\mathit{Value} \sqcap \exists\, \mathit{hasFrequency}.\mathit{Time} \sqcap \exists\, \mathit{hasMethod}.\top$
   $\sqcap \exists\, \mathit{hasResponsable}.(\mathit{Person} \sqcup \mathit{Sensor}) \sqcap \exists\, \mathit{hasVerification}.\mathit{Time} \sqcap$
   $\exists\, \mathit{hasRecord}.\mathit{Document} \sqcap \exists\, \mathit{hasCorrective}.\mathit{Action}$
10. $\mathit{PotentialHazard} \equiv \mathit{Hazard} \sqcap \exists\, \mathit{identifiedIn}.\mathit{ProcessingStep}\, \exists\, \mathit{has}.\mathit{ControlMeasure}$
11. $\mathit{SignificantHazard} \equiv \mathit{Hazard} \sqcap \exists\, \mathit{has}.\mathit{Justification} \sqcap \exists\, \mathit{has}.\mathit{ControlMeasure}$
12. $\mathit{NonSignificantHazard} \equiv \neg \mathit{SignificantHazard} \sqcap \mathit{has}.\mathit{Justification}$
13. $\mathit{SignificantPotentialHazard} \equiv \mathit{PotentialHazard} \sqcap \mathit{SignificantHazard}$
14. $\mathit{CCP} \equiv \mathit{ProcessingStep} \sqcap \exists\, \mathit{has}.\mathit{ControlMeasure}$
15. $\mathit{salmonella} : \mathit{BacterialContamination}$
16. $\mathit{washHands} : \mathit{ControlMeasure}$

Figure 8.5: Top level of the hazards ontology Θ'.

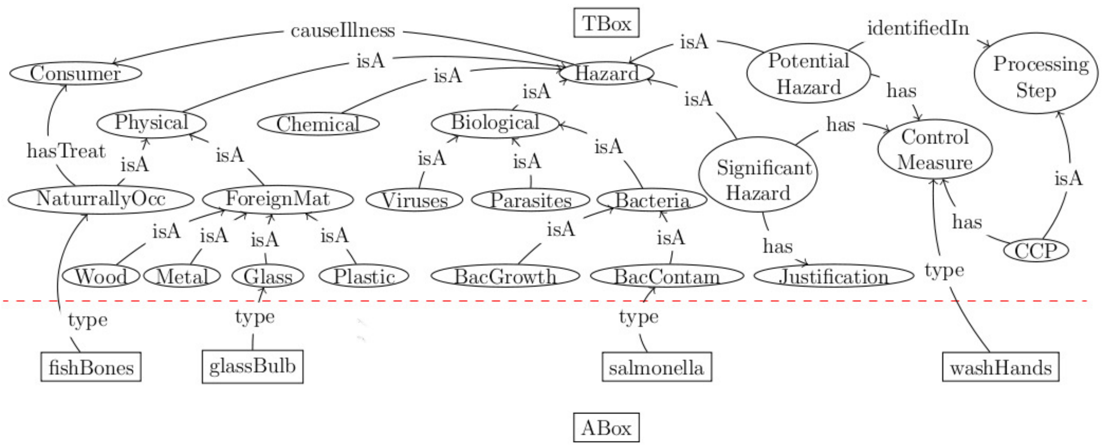


Figure 8.6: Graphical view of the top level hazard ontology Θ'.

(i.e. bones in fish, represented as assertional box in line 5) that are a threat to the consumer [12]. Common sources found in food processing facilities are light bulbs (line 7). A control measure should specify a critical limit, should be performed based on a predefined schedule by a person or a device, and should also be associated to a corrective action in case the critical limit is violated (line 8).

The analysis starts by identifying potential hazards in each step-point in the processing flow (line 10). During the hazard analysis the significance of each hazard is assessed, based on the likelihood of occurrence and severity. The decision is taken from experience, epidemiological data and possible consequences of each hazard. Difference of opinions, even among experts, may occur regarding the significance of a hazard, but each decision should have a clear justification as part of an HACCP plan (given by $\exists\, has.Justification$ in line 11). In an HACCP implementation, any hazard that was declared significant should have at least one control measure to keep it within the critical limits (given by $\exists\, has.ControlMeasure$ in line 11). The disjoint category of hazards, denoted by the concept *NonSignificantHazards* in line 12, should also be supported by a justification in the HACCP documentation. There is no need to define control measures for hazards that are not significant. Furthermore, a critical control point is defined as a processing step in which specific control measures should be applied to prevent the occurrence of a potential significant hazard (line 14).

Part of the HACCP ontology is shown in figure 8.6, where the top level illustrates the axioms in $TBox$, and the bottom part exemplifies a possible $ABox$. The instance *fishBones* is a physical hazard of type naturally occurring object which has some threat to a possible consumer. A glass bulb is an instance of the *Glass* concept, which is physical hazard of type foreign material. *Salmonella* is an instance of bacterial contamination, which is a sub-concept of bacterial-type biological hazard. For each potential hazard a control measure, such as the instance *washHands*, should be specified.

Additionally to the HACCP ontology, we also need the domain knowledge Θ'', that encapsulates the specifics of the business process selected to be modeled. The shrimp ontology integrates knowledge from the shrimp, health, and legal domain. Firstly, the compliance checking agents should have strong knowledge about the current business domain, like shrimp classification, ingredients and additive effects on the processing steps. A short example is illustrated in the figure 8.7, line 31.

Secondly, the agents should have knowledge related to possible food-generated diseases, pathogen taxonomies, possible causes, possible symptoms, allergies, or side-effects. The knowledge extracted from the medical sources is generally oriented towards symptoms or disease definitions (line 33). In the context of food safety, it is better to reformulate the definition centered on hazard terms: sulfites, a chemical hazard, can cause breathing difficulty, or migraines at a population

31. $MantisShrimp \sqsubseteq Shrimp$, $MysidShrimp \sqsubseteq Shrimp$
32. $Sulfites \sqsubseteq Chemical$, $BreathingDifficulty \sqsubseteq AllergicReaction$
33. $BreathingDifficulty \sqsubseteq \forall\, isCausedBy(... \sqcup Sulfites \sqcup ...)$
34. $isCausedBy^{-1} \equiv canCause$
35. $Sulfites \sqsubseteq \forall\, canCause.(BreathingDifficulty \sqcup Migraines \sqcup SkinRash \sqcup Itching \sqcup Flushing$
$\sqcup Tingling \sqcup Swelling) \sqcap \forall\, hasRiskFor.(Asthmatics \sqcup SalicylateSensitivityPersons)$
36. $ContactIce \equiv \exists\, contact.Food \sqcap \forall\, madeFrom.PotableWater$
37. $ContactSteam \sqsubseteq \forall\, contains.\neg Hazard$
45. $PotableWater \sqsubseteq CleanWater$
46. $SalmonellaGr \sqsubseteq \exists\, hasAw. > 0.94 \sqcap \exists\, hasPh.[3.7, 9.5] \sqcap \exists\, hasSalt. < 8\% \sqcap \exists\, hasTemp.[5.2, 46.2]$
47 $SalmonellaGr \sqsubseteq PossBacterialGrowth$
51. $Fresh \sqsubseteq \forall\, hadFishingTime. < 48h$
52. $Frozen \sqsubseteq \forall\, hasTemperature. < 0C$
67. $FrozenStorage \sqsubseteq Storage \sqcap Frozen$
68. $Thawing \sqsubseteq \exists\, contact.Fish$
71. $Shrimp \sqsubseteq Food$
72. $Sulfites \sqsubseteq Preservatives$
73. $washes \sqsubseteq contact$
81. $InadequateCookingTime \sqsubseteq CriticalLimitViolation$
82. $InadequateCookingTime \sqsubseteq \exists\, hasCookingTime. < 1minute$
83. $HoldAndEvaluate \sqsubseteq ControlMeasure$, $Recook \sqsubseteq ControlMeasure$
84. $Label \sqsubseteq ControlMeasure$
85. $bisulfite : Sulfites$
86. $sh1 : MysidShrimp$
87. $64h : > 48Hours$
88. $spain : Country$

Figure 8.7: Shrimp-related hazard ontology Θ''.

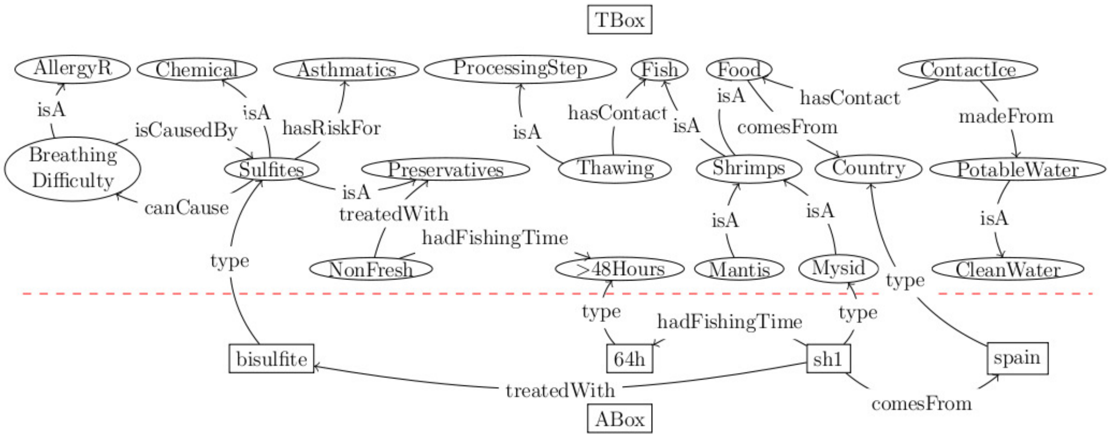


Figure 8.8: Graphical view of the domain specific ontology Θ''.

with increased risk like asthmatics and salicylate sensitivity persons (line 35).

Thirdly, business entities should obey the current regulations and be aware of the legal consequences of their actions[2]. Two examples of such norms are: "Ice which comes into contact with food or which may contaminate food is to be made from potable water" (line 36), and "Steam used directly in contact with food is not to contain any substance that presents a hazard to health or is likely to contaminate the food" (line 37). Parameters defining the conditions for *SalmonellaGrowth* are specified in line 46.

Part of the shrimp domain ontology is graphically depicted in figure 8.8. The individual *sh1* is an instance of mysid subtype of shrimps. It was fished 64 hours ago, value which belongs to the concept $>$ *48Hours*. Based on the terminology in *TBox*, one can infer that $sh1$ is a *NonFresh* item. Consequently, it should be treated with a kind of preservative. This holds in the example because *sh1* is treated with *bisulfite* which is of type *Sulfites*, and thus a preservative too.

8.5.2 Semantic Kripke structure

Running scenario. Consider the following processing workflow. Frozen raw shrimp is received from international and domestic sources. The shrimps from international sources are kept in frozen units. The buying requirement specifies that the shrimp must not contain any sulfite residual and a certification attesting the absence of sulfites should accompany the items. Fresh shrimps are acquired

[2]Food safety norms can be extracted from the EU directive about fishery, Regulation no 852/2004 of the European Parliament, Regulation no 178/2002 about traceability, Fish and Fisheries Products Hazards and Controls Guidance.

61. $Receiving \sqsubseteq ProcessingStep$
62. $Storage \sqsubseteq ProcessingStep$
63. $Thawing \sqsubseteq ProcessingStep$, $Processing \sqsubseteq ProcessingStep$
64. $Cooking \sqsubseteq ProcessingStep$, $Drying \sqsubseteq ProcessingStep$
65. $Packing \sqsubseteq ProcessingStep$, $Delivering \sqsubseteq ProcessingStep$
91. $r_1 : Receiving, r_2 : Receiving, r_3 : Receiving, r_4 : Receiving$
92. $s_1 : Storage, s_2 : Storage, s_3 : Storage$
93. $t_1 : Thawing, t_2 : Thawing$
94. $p_1 : Processing$
95. $c_1 : Cooking, c_2 : Cooking$
96. $d_1 : Drying, pa : Packing, de : Delivery$
101. $(r_1, s_1) : hasNextState, (r_2, s_1) : hasNextState, (r_3\ t_1) : hasNextState$
102. $(r_3\ t_2) : hasNextState, (r_3, s_2) : hasNextState, (r_4, t_1) : hasNextState$
103. $(r_4, t_2) : hasNextState, (r_4, s_2) : hasNextState, (s_1, t_1) : hasNextState$
104. $(s_1, t_2) : hasNextState, (s_2, t_1) : hasNextState, (s_2, t_2) : hasNextState$
105. $(t_1, p_1) : hasNextState, (t_2, t_1) : hasNextState, (t_2, p_1) : hasNextState$
106. $(p_1, s_3) : hasNextState, (s_3, c_2) : hasNextState, (s_3, c_1) : hasNextState$
107. $(c_1, d_2) : hasNextState, (c_2, d_1) : hasNextState, (d_1, pa) : hasNextState$
108. $(pa, de) : hasNextState$

Figure 8.9: Description of the business process in $\mathcal{ALC}$.

directly from local fishermen. The shrimps are often treated with sulfiting agents such as sodium bisulfite or sodium metabisulfite to prevent black spot formation. Shrimps are mixed with ice into recipients containing potable water. Before cooking, defreezing is achieved by passing the product through the thawing process. The thawing uses potable water maintained at 18 °C to 33 °C and circulated with aeration. The finished product is conveyed to the label station, where an automatic system weighs the shrimp and bags the correct quantity in pre-labeled bagging material.

Converting the business process. In a first step, knowledge representing the business workflow is used to generate the corresponding Kripke structure. The eight processing steps identified in the above shrimp scenario: receiving, storage, thawing, processing, drying, cooking, packing, and delivery are shown in the figure 8.9. Several states may be needed to model a single processing step. For instance, the states r_1, r_2, r_3, and r_4 are all instances of the concept *Receiving* (line 91). The processing flow is modeled by the role *hasNextState* which has as domain and range *ProcessingStep* concepts and the inverse role *hasPastStep*. The starting states are automatically identified based on $NotStartState \equiv ProcessingStep \sqcap \exists\, hasPastStep.ProcessingStep$. The ontology in figure 8.9 is graphically illustrated by figure 8.10. As an example, the individual r_1 of type *Receiving* processing step is related with the *Storage* processing step s_1 through the role *hNS*, where *hNS* stands for *hasNextState*. The resulting Kripke

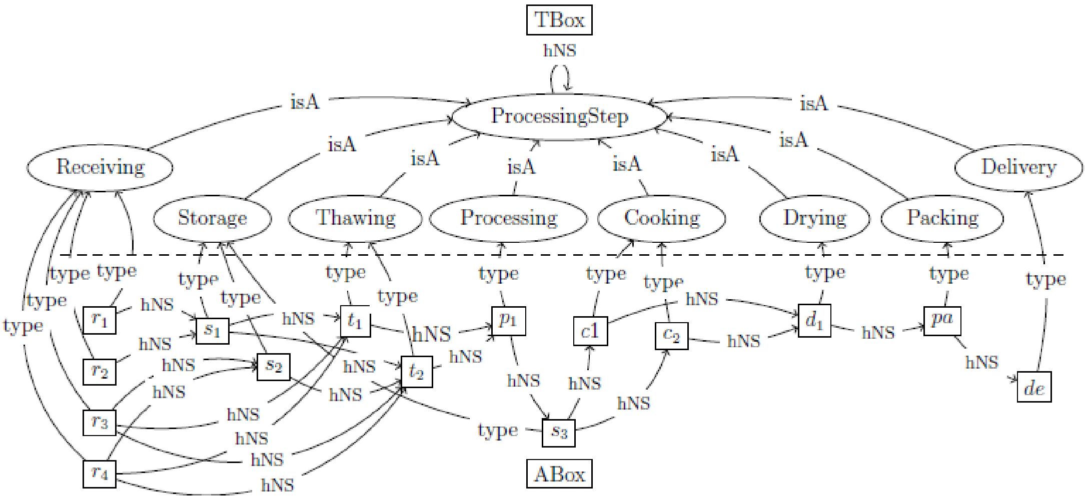


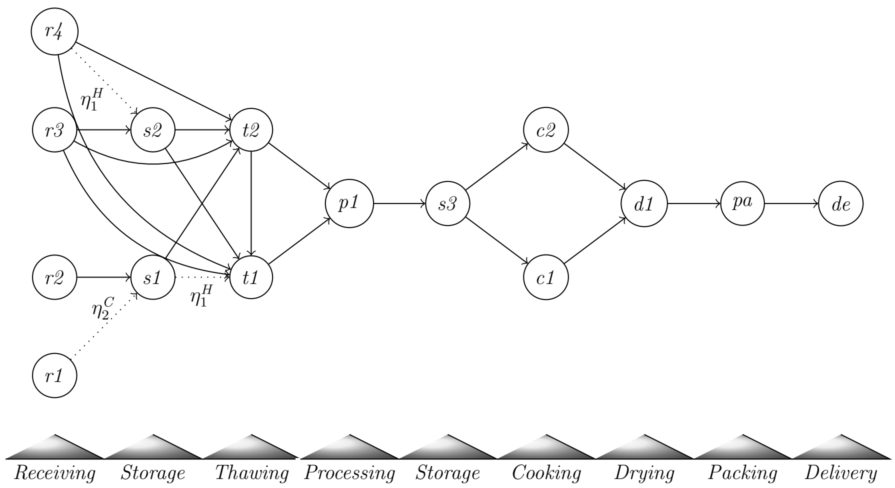
Figure 8.10: Graphical view of the business process ontology.

structure is depicted in the top level of figure 8.11, while the remaining of the figure enumerates the individuals and measured values in each state, representing the domain of interpretation $\Delta^{I(s)}$ for that state.

The state r_1 models the lot sh_1 of mantis shrimps $MantisShrimp^{I(r_1)} = \{sh1\}$ which, base on $hasTemperature(sh1, -20\ ^{circ}C)$ and axiom 52, comes frozen from the international source Thailand ($InternationalSource^{I(r_1)} = \{thai\}$). The lot has provenance certificate *cert1*, and there exists the possibility to be contaminated with bisulfites ($Sulfites^{I(r_1)} = \{possBisulf\}$). The difference between r_1 and r_2 is given by the existence of the supplementary certificate *cert3* of no preservatives for the shrimps *sh2* received in r_2.

Each state is characterized directly by the concepts which are satisfied in that state, and not by the attribute values. Instead of using separate states for the shrimps having the temperatures $-15, -16, -17 ...^\circ$ C, a single, more general state suffices, in which the *Frozen* concept is satisfied. Both states r_3 and r_4 model the shrimps received from the same domestic source $DomesticSource^{I(r_3)} = DomesticSource^{I(r_4)} = \{spain\}$. The fishing time makes the difference by satisfying the concept *Fresh* in the state r_3, justified by axiom 51, but satisfying the concept *NotFresh* in r_4.

The frozen shrimp from international sources (states r_1 and r_2) is stored in freezing units (s_1). Sensors monitoring attributes like pH, temperature, and salt concentration are needed. The fresh shrimp from domestic sources may be short-time stored in non frozen units (s_2) or it may go directly to the thawing steps t_1 and t_2. The shrimp lots *sh3* and *sh4* are stored at $5-6^\circ$ C in state s_2, where the identified



$\Delta^{I(r_1)} = \{sh1, thai, cert1, possBiSulf, -20\,^\circ C\}$ $comesFrom^{I(r_1)}\{(sh1, thai)\}$ $hasTemperature^{I(r_1)} = \{(sh1, -20\,^\circ C)\}$ $hasCertificate^{I(r_1)} = \{(sh1, cert1)\}$ $ProvenanceCertificate^{I(r_1)} = \{cert1\}$ $MantisShrimp^{I(r_1)} = \{sh1\}$ $InternationalSource^{I(r_1)} = \{thai\}$ $Sulfites^{I(r_1)} = \{possBisulf\}$	$\Delta^{I(r_2)} = \{sh2, vietnam, cert2, cert3, -18\,^\circ C\}$ $comesFrom^{I(r_2)} = \{(sh2, vietnam)\}$ $hasTemperature^{I(r_2)} = \{(sh2, -18\,^\circ C)\}$ $hasCertificate^{I(r_2)} = \{(sh2, cert2), (sh2, cert3)\}$ $ProvenanceCertificate^{I(r_2)} = \{cert2\}$ $MySidShrimp^{I(r_2)} = \{sh2\}$ $InternationalSource^{I(r_2)} = \{vietnam\}$ $NoPreservativesCertificate^{I(r_2)} = \{cert3\}$
$\Delta^{I(r_3)} = \{sh3, spain, possSal, 36h\}$ $comesFrom^{I(r_3)} = \{(sh3, spain)\}$ $hadFishingTime^{I(r_3)} = \{(sh3, 36h)\}$ $MySidShrimp^{I(r_3)} = \{sh3\}$ $DomesticSource^{I(r_3)} = \{spain\}$ $BacterialContamination^{I(r_3)} = \{possSal\}$	$\Delta^{I(r_4)} = \{sh4, spain, possSal, 52h\}$ $comesFrom^{I(r_4)} = \{(sh4, spain)\}$ $hadFishingTime^{I(r_4)} = \{(sh4, 52h)\}$ $MySidShrimp^{I(r_4)} = \{sh4\}$ $DomesticSource^{I(r_4)} = \{spain\}$ $BacterialContamination^{I(r_4)} = \{possSal\}$
$\Delta^{I(s_1)} = \{sh1, sh2, -20\,^\circ C, -18\,^\circ C\}$ $hasTemperature^{I(s_1)} = \{(sh1, -20\,^\circ C), (sh2, -18\,^\circ C)\}$	$\Delta^{I(s_2)} = \{sh3, sh4, possSalmonella, 6\,^\circ C, 5\,^\circ C\}$ $hasTemperature^{I(s_1)} = \{(sh3, 6\,^\circ C), (sh4, 5\,^\circ C)\}$ $PossBacterialGrowth^{I(s_1)} = \{possSalmonella\}$
$\Delta^{I(t_1)} = \{w1, 9.12\}$ $hasPh^{I(t_1)} = \{(w1, 9.12)\}$	$\Delta^{I(t_2)} = \{w2, 8.65\}$ $hasPh^{I(t_2)} = \{(w2, 8.65)\}$
$\Delta^{I(p_1)} = \{bacillusCereus\}$ $BacterialGrowth^{I(p_1)} = \{bacillusCereus\}$	$\Delta^{I(s_3)} = \{possSal1\}$ $BacterialGrowth^{I(s_3)} = \{possSal1\}$
$\Delta^{I(c_1)} = \{sh1, oil, 1.3minutes, 378\,^\circ C\}$ $hasCookingTime^{I(c_1)} = \{(sh1, 1.3minutes)\}$ $hasTemperature^{I(c_1)} = \{(oil, 378\,^\circ C)\}$	$\Delta^{I(c_2)} = \{sh2, oil, possSal2, 0.8minutes\}$ $hasCookingTime^{I(c_2)} = \{(sh2, 0.8minutes)\}$ $PathogenSurvival^{I(c_2)} = \{possSal2\}$
$\Delta^{I(d_1)} = \{possSal3\}$ $BacterialGrowth^{I(d_1)} = \{possSal3\}$	$\Delta^{I(pa)} = \{staphylococcus\}$ $PathogenContamination^{I(pa)} = \{staphylococcus\}$

Figure 8.11: Refined business process for shrimp processing.

Table 8.2: Labeling states.

State	*State-related labels*	*Concept-related labels*
r1	Receiving, PS, SS	MantisShrimp, Shrimp, Food, Frozen, Sulfites, Preservatives, Chemical, Hazard, InternationalSource, ProvenanceCertificate
r2	Receiving, PS, SS	MySidShrimp, Shrimp, Food, Frozen, InternationalSource, ProvenanceCertificate, NoPreservativeCertificate
r3	Receiving, PS, SS	MySidShrimp, Shrimp, Food, Fresh, DomesticSource, BacterialContamination, Bacterial, Hazard
r4	Receiving, PS, SS	MySidShrimp, Shrimp, Food, NonFresh, DomesticSource, BacterialContamination, Bacterial, Hazard
s1	Storage, PS, NSS	MantisShrimp, MySidShrimp, Shrimp, Food, Frozen, FrozenStorage
s2	Storage, PS, SS	MySidShrimp, Shrimp, Food, NonFrozen
t1	Thawing, PS, SS	DirtyWater
t2	Thawing, PS, SS	CleanWater
p1	Processing, PS, SS	BacterialGrowth, Bacterial, Hazard
s3	Storage, PS, SS	BacterialGrowth, Bacterial, Hazard
c1	Cooking, PS, SS	
c2	Cooking, PS, SS	InadequateCookingTime, CriticalLimitViolation
d1	Drying, PS, SS	BacterialGrowth, Bacterial, Hazard
pa	Packing, PS, SS	PathogenContamination, Hazard
de	Delivery, PS, SS	

hazard "possibility of salmonella growth" satisfies the concept *PossBacterialGrowth* due to the axioms 46 and 47. In t_2, fresh potable water is used, while in t_1 the water is recirculated. Note that different normative systems define differently the concept of potable water: the EU recommendation Directive 98/83/EC of 1998 specifies the open interval (6.5,9.5) for the pH value, while the Denmark value[3] should belong to the closed interval [7.0,8.5]. Thus, the concept *CleanWater* is satisfied in t_2 only under the EU legal water ontology.

For the processing step p_1 and the storage step s_3, the biological hazard *BacterialGrowth* is satisfied. No hazard exists in the cooking step c_1, because adequate cooking conditions are used, while the fast cooking of only 0.8 minutes in c_2 leads to the possibility of pathogen survival. The *InadequateCookingTime* concept being satisfied in c_2 (line 82), leads to a violation of the critical limit, according to line 81 in Fig. 8.7. If the drying is not executed properly in the state d_1, the remaining wet spots are ideal places for bacteria growth, while contamination with staphylococcus from the packing material can occur in the state pa.

Applying algorithm 4, each state is labeled with all satisfied concepts. Table 8.2 bears out both knowledge describing states and domain dependent concepts. Here, *PS* stands for *ProcessingStep*, *SS* for *StartState*, and *NSS* for *NotStartState*. For

[3]Statutory order no 871 of 21 September 2001 on water quality and inspection of water supply plants issued by Danish Ministry of Environment.

$\eta_1^H \langle \mathit{NotFresh} \sqcap \mathit{DomesticSource}, \mathit{Storage} \rangle \quad (r_4, s_2)$
$\eta_2^C \langle \mathit{InternationalSource} \sqcap \forall\, \mathit{hasCertificate}.\neg \mathit{NoSulfites}, \mathit{ProcessingStep} \rangle \quad (r_1, s_1)$
$\eta_4^H \langle \mathit{FrozenStorage}, \mathit{Thawing} \sqcap \neg \mathit{CleanWater} \rangle \quad (s_1, t_1)$

Figure 8.12: Active norms in the business process.

instance, the individual r_1 is classified as a *Receiving* state, as a *ProcessingStep* and as a *StartState*. Subsumption chains like $\mathit{MantisShrimp} \sqsubseteq \mathit{Shrimp} \sqsubseteq \mathit{Frozen}$ or $\mathit{Sulfites} \sqsubseteq \mathit{Preservatives} \sqsubseteq \mathit{Chemical} \sqsubseteq \mathit{Hazard}$ are added to the r_1 state labels.

8.5.3 Enacting the norms

In the third step the norms are applied to the business process. Three norms are exemplified in Fig. 8.12. The norm η_1^H states that "Only fresh shrimps can be stored". The concept *NotFresh* is satisfied in r_4, where the shrimps coming from Spain satisfy the *DomesticSource* concept. Based on the norm η_1^H the shrimps *sh4* cannot be stored. Consequently all the links between r_4 and the states where the *Storage* concept is satisfied (s_1, s_2, s_3) should be deleted. The transition between r_4 and s_2 is removed, illustrated by a dotted line in Fig. 8.11.

The norm "For shrimps from international sources a certification should attest that they do not contain any sulfites" is represented by η_2^C and regards the states r_1 and r_2 where the concept *InternationalSource* is satisfied. A certificate of no preservatives exists in r_2 satisfying this norm based on the subsumption $\mathit{Sulfites} \sqsubseteq \mathit{Preservatives}$ with the following subsumption:

$$\forall\, \mathit{hasCertificate}.\mathit{Sulfites} \sqsubseteq \forall\, \mathit{hasCertificate}.\mathit{Preservatives}$$

Because no certificate regarding sulfites or more general additives exists in state r_1, all the links between r_1 and any processing step are removed. In this case s_1 is a storage step, where $\mathit{Storage} \sqsubseteq \mathit{ProcessingStep}$.

The norm η_4^H, "Frozen stored shrimps should be thawed in clean water", refers to the state s_1, where *FrozenStorage* is satisfied based on axiom 67 in Fig. 8.7. The *CleanWater* concept is not satisfied in t_1, therefore the relation (s_1, t_1) is removed from R. Now assume that the more restrictive norm $n_4^H \sqsubseteq \eta_1^E$ is issued by the government body: "Water which comes in contact with food should be potable":

$$\eta_1^E \langle \neg \mathit{PotableWater} \sqcap \exists\, \mathit{contact}.\mathit{Food}, \mathit{ProcessingStep} \rangle$$

The subsumption between the two norms is based on the following inference chain:

$\overset{\mathit{lines}\ 63,68}{\rightarrow} \quad \mathit{Thawing} \sqsubseteq \mathit{ProcessingStep} \sqcap \exists\, \mathit{washes}.\mathit{Shrimp}$
$\overset{\mathit{line}\ 71}{\rightarrow} \quad \mathit{Thawing} \sqsubseteq \mathit{ProcessingStep} \sqcap \exists\, \mathit{washes}.\mathit{Food}$
$\overset{\mathit{line}\ 73}{\rightarrow} \quad \mathit{Thawing} \sqsubseteq \mathit{ProcessingStep} \sqcap \exists\, \mathit{contact}.\mathit{Food}$

and from the definition in line 45, it follows that $\neg \mathit{PotableWater} \sqsupseteq \neg \mathit{CleanWater}$.

8.5.4 Verifying regulatory compliance

Four compliance patterns (*CPs*) are formalized in $\mathcal{NTL} - \mathcal{ALC}$, where the index H stands for the HACCP standard.

CP 1 *According to the current requirements about traceability "Shrimp from international sources should have a certification of provenance", expressed by the following norm.*

$$O_H\Box(InternationalSource \Rightarrow O_H\,\exists\, hasCertificate.ProvenienceCertificate)$$

In Table 8.2, the concept *InternationalSource* is satisfied in the states r_1 and r_2, where the role *hasProvenanceCertificate* exists in both states. The consequent formula in the implication does not include any temporal operator, meaning that the concept occurring in the conclusion should be satisfied in the same state as the premise. Consequently, the compliance pattern CP 1 is satisfied.

CP 2 *According to the HACCP, at each moment of time when a deviation from the critical limit occurs, a control measure should be applied in the very next state.*

$$O_H\Box(CriticalLimitViolation \Rightarrow O_H \odot ControlMeasure)$$

Because $InadequateCookingTime \sqsubseteq CriticalLimitViolation$, the deviation from a critical limit has occurred in state c_2 (see table 8.2), meaning that in the next state one has to activate a specific control measure. Two such control measures are defined in the domain ontology Θ'': one has either to hold and evaluate the items or to recook, as specified in the line 83 of figure 8.7. According to the labels in the table 8.2 assigned to the only next state d_1, the concept *ControlMeasure* is not satisfied. Thus, the compliance pattern CP2 is violated.

CP 3 *At each moment of time when a preservative is included in an item, one has the obligation to label it accordingly before delivery to the retailer.*

$$O_H\Box(Preservatives \Rightarrow O_H\Diamond Label)$$

Assume that sulfites, a particular class of preservatives given by $Sulfites \sqsubseteq Preservatives$, are added to the shrimps to avoid black spot formation. As table 8.2 bears out, the *Label* concept, acting as a control measure in the HAACP plan ($Label \sqsubseteq ControlMeasure$), is satisfied in no state of the Kripke structure.

CP 4 *The product shall be kept frozen to maintain the quality during delivery.*

$$O_H\Box(Delivery \Rightarrow O_H\Box Frozen)$$

The labeling algorithm also attaches the type of the processing step (i.e. *Receiving*, *Storage*) to the current state. The increased expressiveness given by this property, allows the auditor to define patterns like CP4 which refers in its premise to the *Delivery* states. A negative answer to this query can help the decision maker to include missing relevant information in the logs or labels. For instance, assume that *Frozen* is not satisfied in the *Delivery* state, due to the fact that the temperature is not specified for that particular state. The missing information related to temperature, which is relevant in this particular case, will be included in the Kripke model and in the HACCP documentation, to guarantee the CP4 property.

8.6 Conclusions

We have introduced the $\mathcal{NTL} - \mathcal{ALC}$ logic aiming at checking norm compliance in business processes. Combining domain knowledge with model checking, an enhanced model checker has been developed by integrating the ACL2 theorem prover with the reasoning services of Racer. The main benefit of integrating subsumption-based reasoning with model checking is the possibility to check the norms against a range of concepts, from abstract to more specific ones, as identified in ontologies. The $\mathcal{NTL} - \mathcal{ALC}$ logic provides therefore a generic framework to close the gap between abstract norms and concrete business processes.

We applied $\mathcal{NTL} - \mathcal{ALC}$ to the HACCP quality standard for a realistic scenario. By analyzing the requirements for supporting an HACCP-based system, we have shown the need to introduce domain knowledge when checking, for helping a third party, like an auditor, to verify compliance, and also to increase transparency. Quite aware of the difficulty of running a realistic scenario, we have developed three ontologies for representing HACCP related concepts, for encapsulating specific business knowledge, and another for supporting the conversion of a business workflow description into a Kripke model. By changing the ontologies or importing new ones, the framework can be applied to various normative systems.

The efficiency problems associated to model checking are under control since checking business processes is quite different compared to checking software applications. In practice, most business processes have a relatively small state space [41]. Therefore, the complexity is given not by the size of the model, but rather by the number of regulatory obligations that a particular process should comply with. Techniques like abstraction and segmentation can be applied simply by using an adequate amount of domain knowledge for the specific problem. Similarly, the expressiveness is mainly supported by the version of description logic used. We have used here the $\mathcal{ALC}$ logic, but more expressive versions can be useful when modeling complex scenarios.

Part III

Future work

Chapter 9

Overall plans for the future

> "Daca nu stii unde mergi o sa ajungi altundeva"
>
> Yogi Berra

9.1 Teaching Artificial Intelligence

I am currently working on two didactic books: The first one is on teaching logic with puzzles. The second one is on difficulties of human reasoning with probabilities.

The content of these books owes much to the cultural and pleasant atmosphere at the Intelligent Systems Group (ISG). Many examples were stimulated by the discussions with Radu Razvan Slavescu and Anca Marginean. Radu has always manifested his ability to extract general principles for each specific example. His endless cultural references have been always a stimulus for Anca and I to think out of the box. Anca's empathy and her tact to motivate and encourage students have been always an example to me. Her teaching model has triggered endless debates within the ISG on how to approach the student: either a left-strategy (call it Che Guevara strategy - which approaches the student as a comrade and empathizes with the students difficulties etc.), or a far-right strategy (call-it dictatorship that firmly guides the students throughout the intriguing and vast world of AI).

9.1.1 Teaching logic through puzzles

I am interesting to compare the ability of human agents and logical agents to solve puzzles [96, 90, 123]. Examples of puzzles (taken from [96]) that can be solved by theorem provers (like Prover9) and model finders (like Mace4) are:

Puzzle 1 *"Little indian and big indian are walking side by side. Big indian is the father of little indian, but little indian isn't the son of the big indian. How is this possible?"*

Puzzle 2 *"Two girls are born to the same mother, on the same day, in the same month and year and yet they're not twins. How can this be?"*

Puzzle 3 *"Komsomol youths have built a small hydroelectric powerhouse. Preparing for its opening, young communist boys and girls are decorating the powerhouse on all four sides with garlands, electric bulbs, and small flags. There are 12 flags. At first they arrange the flags 4 to a side, as shown, but then they see that the flags can be arranged 5 or even 6 to a side. How?"*

Puzzle 4 *"It is easy to arrange 16 checkers in 10 rows of 4 checkers each, but harder to arrange 9 checkers in 6 rows of 3 checkers each. Do both."*

Puzzle 5 *"There are three ways to add four odd numbers and get 10:*

$$1 + 1 + 3 + 5 = 10$$
$$1 + 1 + 1 + 7 = 10$$
$$1 + 3 + 3 + 3 = 10$$

Changes in the order of numbers do not count as new solutions. Now add eight odd numbers to get 20. To find all eleven solutions you will need to be systematic".

```
set(arithmetic).
assign(domain_size, 15).
assign(max_models, -1).
formulas(assumptions).

odd(x) <-> x mod 2 = 1.
odd(A) & odd(B) & odd(C) & odd(D) & odd(E) & odd(F) & odd(G) & odd(H).
A + B + C + D + E + F + G + H = 20.
A >= B & B >= C & C >= D & D >= E & E >= F & F >= G & G >= H & H > 0.

end_of_list.
```

Mace4 has no difficulties to identify satisfiable models for the constraints given (see Table 9.1).

9.1.2 Lecture notes on probabilistic reasoning

Each year when I teach the six classes on Bayes nets and rational decisions, I found myself in the position of finding embarrassing gaps in my understanding and

Table 9.1: 11 models for puzzle p5.

Model	A	B	C	D	E	F	G	H
1	3	3	3	3	3	3	1	1
2	5	3	3	3	3	1	1	1
3	5	5	3	3	1	1	1	1
4	5	5	5	1	1	1	1	1
5	7	3	3	3	1	1	1	1
6	7	5	3	1	1	1	1	1
7	7	7	1	1	1	1	1	1
8	9	3	3	1	1	1	1	1
9	9	5	1	1	1	1	1	1
10	11	3	1	1	1	1	1	1
11	13	1	1	1	1	1	1	1

annoying hesitations when solving exercises involving probabilities. So I decided to make a virtue out of the necessity, by elevating my struggle with probabilities to the level of one booklet on probabilistic reasoning.

This book will be written as a discussion between several characters: a teacher, an eager student, a lazy student, a logician, a cognitive scientist and an encyclopedist. The source of inspiration for this format was a class hold by Jan van Eijck and Rineke Verbrugge at the ESSLLI summer school in Hamburg in 2008 (or Hans van Ditmarsch and Jan van Eijck). Quickly, I read their edited book *Discourses on Social Software* [40]. The book was the main support to design some highly interactive classes on epistemic logic at Technical University of Cluj-Napoca. Of course, there are several textbooks following the same discussion style. In 1632, Galileo Galilei compared the Ptolemy and Copernican world views in his book *Dialogue Concerning the Two Chief World Systems* [46]. Raymond Smullyan used dialogues *The Tao is silent* [156]. Not to mention the Hofstader's *Godel, Escher, Bach* [82]. Plato also used fictive conversations between Socrates and his contemporaries to put his ideas forward. Touretzky relies on funny stories to introduce recursion [163]. A lazy dragon helps through dialogue an eager apprentice to figure himself out concepts related to recursion. Touretzky is also aware of the risk of such light stories: while students enjoy and find them helpful, professors have conflicting views on such teaching techniques.

I argue that presenting lectures as fictive dialogues is a natural way to meet part of the needs of the modern student:

First, one pedagogical tip says "Never say something to students that they can say themselves." Indeed, given a class of 100 heterogeneous students, there is a high chance that most of questions addressed to them to find an answer from the audience. Not to mention that a student recalls easier a piece of information conveyed by one of his or her peers. The dialogues in this book follow this pedagogical advice. Whenever possible, the arguments and explanations come not from the teacher but from the students themselves.

Second, teachers like a top-down approach, moving from abstract principles and definitions towards more concrete examples. Differently, students prefer a bottom-up approach, so they like to begin with examples, from which general principles can then be derived[1].

Third, the student is demanding for a *standup teacher*. The imperative of fun. As knowledge is everywhere online, the burden during classes is to provide motivation and make class time fun.

By expanding the teaching styles used in computer science classrooms, we can accommodate the heterogeneity of students, aiming to promote an active learning environment, Pollard and Duvall argue on the role of such dialogue style to expand the audience in computer science to students from other fields [143]. Presentation in dialogue form is more effective than monologue [152, 102]. Craig et al. have found that dialogue stimulates students to write more in a free recall test and ask twice as many questions [152]. Lee et al. have reported that there is more discussion between students and less banter after watching a dialogue [102]. The effectiveness of dialogue over monologue is also witnessed by the widespread use of dialogue in commercials or news bulletins.

An excerpt from the working version of the book follows:

[1] Unfortunately, given the poor time management skills of both teachers and students, teachers spend most of the lectures on definitions and only last minutes with examples, while students use all their time to cover exams from previous years, and delaying the learning of general principles for good.

TEACHER: "Suppose you're a hero that wants to save a princess from an evil dragon. The dragon - call it Monty Hall - gives you the choice of three doors in his prison: behind one door is the princess; behind the others, beastly tigers. If you find the princess you married her and you get half of the kingdom. You pick a door, say Door A [but the door is not opened], and the dragon, who knows what's behind the doors, opens another door, say Door B, and shows to you the tiger inside the cell. He then says to you, "Do you want to pick Door C?" Is it to your advantage to switch your choice?"
ENCYCLOPEDIST: This is an adapted version of the famous Monty Hall problem. It is based on *Let's Make a Deal* TV show. Mathematicians, psychologists, cognitive scientists, philosopher or economists have found various explanations why people find this problem so difficult. For instance, Rosenhouse spent an entire book on this brain teaser problem only [151].
LAZY: Have you seen the move *21* directed by Robert Luketic? The movie was inspired by the MIT Blackjack team who used card counting techniques to beat the casinos. This problem appears in the movie also, if I recall correctly. I'm intrigued about the solution.
LAZY: As far as I am concerned, the two remaining options are equally likely to conceal the princess. as it makes no difference what decision we take, I will stay with my first choice: door A.
LOGICIAN: I wonder how the beast dragon chose his door to open.
COGNITIVE SCIENTIST: It matters only that just two doors remain. Important is that they have equal probabilities of cover the princess!
LOGICIAN: This is cogent logic. There is indeed relevant the procedure followed by the dragon in picking his door.
COGNITIVE SCIENTIST: The Monty Hall problem is a nice illustration of the difficulties most people do have when reasoning with uncertainty.

9.2 Research topics

"We can see only a short distance ahead, but we can see that much remains to be done"

Turing

9.2.1 Explainable AI

I have perfectly resonate with the following qoutation [134] from CACM:

"It's time for AI to move out its adolescent, game-playing phase and take seriously the notions of quality and reliability."

The goal of Explainable AI (XAI) is to support user with the following questions against a black-box systems (i.e. ML): Why did you do that? Why not something else? When do you succeed/fail? When can I trust you? How to I correct an error?

In the same line, EU policy towards data protection also advocates the *right to explanation*. Hence, I was not surprised by the GDPR article 22: *"The right not to be subject to a decision based solely on automated processing."*

The topic of responsible data science [165] is a first example of a blind spot that can benefit from XAI. My thesis here is that ML alone is not enough for decision making. Even if with accurate classification you need a kind of guarantee or reasoning to take decisions.

Further, XAI can be integrated in the larger fight on algorithmic transparency and accountability [1]. Here, the stakeholders of a system should be aware of the possible harm that algorithms can cause. Regulations should protect users that are negatively affected by algorithmically informed decisions. Organisations are responsible for decisions made by the algorithm they use. Systems should explain both the steps performed by the algorithm and the decision taken. How training data was collected should be explained to the stakeholders.

In my view, XAI could be a promising research field for: i) it is a reaction against black box models of most ML; ii) it is supported by EU policy in AI ethics and data protection; iii) it fits the desiderata in AI to overcome its "teenage period". My previous work on arguments and explanation fits perfectly in newly Explainable AI trend. The tests and validation method should be make public.

My view is that sooner or later, legislation will force some of software products to have a mandatory *Explain me!* button.

9.2.2 Differences of discourse understanding between human and software agents

I am interested in the differences between how a human agent and a logic-based software agent interpret a text in natural language. When reading a narrative, the human agent has a single interpretation model. That is the preferred model among the models consistent with the available information. The model is gradually adjusted as the story proceeds. Differently, a logic-based software agent works with a finite set of many models, in the same time. Of most interest is that the number of these models is huge, even for simple narratives. Let the love story between Abelard and Heloise, with the text "Abelard and Heloise are in love". Assume during natural language processing, the statement is interpreted as Abelard is in love and Heloise is in love. The formalisation in First Order Logic is: (A1) $\exists\, x,\ love(abelard, x)$ (A2) $\exists\, x,\ love(heloise, x)$.

How many models does a model generator find for axioms (A1) and (A2)? Using MACE4 [128], with the domain closed to 4 individuals, there are 278,528 models. All these models are equally plausible for the software agent.

To reduce this number, the agent needs to add several constraints. First, the unique name assumption can be added: (A3) $abelard! = heloise$. Still, there are 163,840 models. Second, we assume that the love relation is not narcissistic: (A4) $\forall\, x, \neg love(x, x)$. That leads to 5,120 models. Third, we add the somehow strong constraint that someone can love only one person at a time. That is (A5) $love(x, y) \wedge love(x, z) \rightarrow y = z$. The remaining models are 80. Unfortunately, love is not a symmetric relation. Hence, we cannot add the axiom $\forall\, x, y\ love(x, y) \equiv love(y, x)$. Instead we exploit the fact that some of these models are isomorphic. After removing isomorphic interpretations, we keep 74 non-isomorphic models. Note that there are 2 Skolem constants after converting axioms (A1) and (A2). If we are not interested in the love relations of individuals represented by these constants, we can ignore them. This would result in 17 models.

Some observations follow. First, the order in which we apply the reductions is computationally relevant. For instance, it would be prohibitively to search for isomorphic models in the initial two steps, when there are 278,528 and 163,840 models. Hence, the strategy is to add domain knowledge to the initial narrative discourse, and then to search for the isomorphic structures. Second, which domain knowledge to add is subject to interpretation. For instance, axiom (A5) might be too strong. Third, for some reasoning tasks (e.g. solving lateral thinking puzzles keeping all possible models might be desirable. Fourth, I argue that text models built with machine learning applied on big data, would benefit from some crash diet. In this line, I try to extract as much as we can from each statement, instead of statistically analysing the entire corpus. That is, the model of the story is built bottom-up and not top-down as machine learning does.

Both the human reader and the software agent aim to keep the story more intelligible and tractable. But they apply different reduction strategies. On one hand, humans understand stories by inferring the mental states (e.g. motivations, goals) of the characters, by applying projections of known stories into the target narrative, by extensively using commonsense reasoning [136], or by closing the world as much as possible. On the other hand, logic-based agents reduce the models by formalising discourse representation theories, by adding domain knowledge, or by identifying isomorphisms.

I will also analyse how the number of interpretation models vary as the story evolves. Sentences introducing new objects and relations increase the number of models. Sentences introducing constraints on the existing objects and relations contribute to the removal of some models. Adding domain knowledge also contributes to model removal. One question is how to generate stories that end with a single interpretation model for software agents. Another issue regards the amount of domain knowledge and commonsense knowledge to add, and which reduction strategy is better to keep the number of models computationally feasible.

To sum up, I am interesting to compare the reduction strategies of humans and software agents to keep the discourse more intelligible and tractable.

9.3 Supervising PhDs

> "Always two there are: a Master and an Apprentice"
>
> Yoda

The following are my preliminary notes on supervising PhD. As experience is little, these notes might be refined and even wrong, but I assume some starting guidelines are better than no idea.

They say that building up a research group, with doctoral students and post-docs is rewarding [37]. I am quite aware of the difficulty of building a research group, but also on the benefits. The group would complement the traditional master-apprentice relation with a partner like relation between its members. I imagine an ideal peer support group for a PhD candidate as a group of three to five peers who meet regularly to discuss the content and process of their research projects. One direction to cover this objective is to start a monthly "Seminar on Logic and Reasoning". Various topics will be discussed by supervisor, PhD candidate, visiting fellows, or masters candidate.

There are different dimensions on how to supervise the student. First, in order to advance the project, a Phd. student should benefit from the supervisor's network. I am aware that I need to invest more attention to build a larger network. Second, given the tradition of apprenticeship in many universities, teachers rely too much on their own experience and too little on more general principles. Many supervisers admit that "basically I learn by doing and learn by mistakes" [78]. Thus, it is not surprising that many universities have introduced formal professional development programs for doctoral supervisors [124], for more than 15 years already. In this line, I aware that the costly "learning by mistakes" can be reduced by following some supervision pedagogies [76].

A fast survey on the literature on doctoral supervision has indicated to me that more guidelines have been designed for PhD students and not for the supervisers.

Take for instance the recent work of Brennan [20]. I wonder how the above notes will be confirmed of invalidated by the experiences to come on this road of supervising PhD students. I am also curious and enthusiastic what will be the impact on my own learning of computer science about the practice of PhD supervision [78].

Bibliography

[1] US ACM. Public policy council and acm europe policy committee, 2017. statement on algorithmic transparency and accountability (25 may), 2017.

[2] Thomas Ågotnes, Wiebe van der Hoek, Juan A. Rodríguez-Aguilar, Carles Sierra, and Michael Wooldridge. On the logic of normative systems. In *IJCAI, 2007*, pages 1175–1180, 2007.

[3] Leila Amgoud and Florence Dupin De Saint Cyr. Measures for persuasion dialogs: A preliminary investigation. In *COMMA 2008*, pages 13–24. IOS Press, 2008.

[4] Leila Amgoud and Mathieu Serrurier. Agents that argue and explain classifications. *Autonomous Agents and Multi-Agent Systems*, 16(2):187–209, 2008.

[5] Leila Amgoud and Srdjan Vesic. A formal analysis of the role of argumentation in negotiation dialogues. *Journal of Logic and Computation*, 22(5):957–978, 2012.

[6] Sergei Artemov. Justified common knowledge. *Theoretical Computer Science*, 357(1-3):4 – 22, 2006. Clifford Lectures and the Mathematical Foundations of Programming Semantics.

[7] Sergei Artemov. Justification logic. In *JELIA*, pages 1–4, 2008.

[8] Sergei Artemov. Why do we need Justification Logic? Technical Report TR–2008014, CUNY Ph.D. Program in Computer Science, September 2008.

[9] Arash Bahrammirzaee. A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. *Neural Computing and Applications*, 19(8):1165–1195, 2010.

[10] Radu Balaj and Adrian Groza. Detecting influenza epidemics based on real-time semantic analysis of twitter streams. In Dana Simian, editor, *Modelling and Development of Intelligent Systems, Sibiu, Romania, 2014*, pages 30–39. Lucian Blaga University Press, 2013.

[11] G. Barbur, B. Blaga, and A. Groza. Ontorich-a support tool for semi-automatic ontology enrichment and evaluation. In *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Confer-*

ence on, pages 129–132. IEEE, 2011.

[12] Dane Bernard, Bob Colletet, Don Kraemer, Kathy Hart, Bob Price, Steve Otwell, and Donn Ward. *Hazard Analysis and Critical Control Point Training Curriculum*. National Seafood HACCP Aliance, 1997.

[13] Massimo Bertolini, Antonio Rizzi, and Bevilacqua Maurizio. An alternative approach to HACCP system implementation. *Journal of Food Engineering*, 79:1322–1328, 2007.

[14] Philippe Besnard and Anthony Hunter. Practical first-order argumentation. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 590. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[15] Larry L. Biehl, Lan Zhao, Carol X. Song, and Christopher G. Panza. Cyberinfrastructure for the collaborative development of {U2U} decision support tools. *Climate Risk Management*, pages –, 2016.

[16] Rodrigo Bonacin, Olga Fernanda Nabuco, and Ivo Pierozzi Junior. Ontology models of the impacts of agriculture and climate changes on water resources: Scenarios on interoperability and information recovery. *Future Generation Computer Systems*, 54:423 – 434, 2016.

[17] R. Booth, M. Caminada, M. Podlaszewski, and I. Rahwan. Quantifying disagreement in argument-based reasoning. In *AAMAS*, pages 493–500, 2012.

[18] Constantine Boussalis and Travis G Coan. Text-mining the signals of climate change doubt. *Global Environmental Change*, 36:89–100, 2016.

[19] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[20] Niamh Brennan. 100 phd rules of the game to successfully complete a doctoral dissertation. *Accounting, Auditing & Accountability Journal*, 32(1):364–376, 2019.

[21] K. Budzynska and C. Reed. Speech acts of argumentation: Inference anchors and peripheral cues in dialogue. In *Computational Models of Natural Argument*, 2011.

[22] H. Bunt, J. Alexandersson, J.-W. Choe, A. C. Fang, K. Hasida, V. Petukhova, A. Popescu-Belis, and D. Traum. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), 2012.

[23] Martin W. A. Caminada and Dov M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009.

[24] Calin Cara, Adrian Groza, Sergiu Zaporojan, and Igor Calmicov. Assisting drivers during overtaking using car-2-car communication and multi-agent systems. In *2016 IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 293–299. IEEE, 2016.

[25] Caterina Caracciolo, Armando Stellato, Ahsan Morshed, Gudrun Johannsen, Sachit Rajbhandari, Yves Jaques, and Johannes Keizer. The AGROVOC linked dataset. *Semantic Web*, 4(3):341–348, 2013.

[26] E. Chaniotakis, C. Antoniou, and F. Pereira. Mapping social media for transportation studies. *IEEE Intelligent Systems*, 31(6):64–70, Nov 2016.

[27] George Chatzieleftheriou, Borzoo Bonakdarpour, ScottA. Smolka, and Panagiotis Katsaros. Abstract model repair. In AlwynE. Goodloe and Suzette Person, editors, *NASA Formal Methods*, volume 7226 of *Lecture Notes in Computer Science*, pages 341–355. Springer Berlin Heidelberg, 2012.

[28] Carlos Chesnevar, Sanjay Modgil, Iyad Rahwan, Chris Reed, Guillermo Simari, Matthew South, Gerard Vreeswijk, Steven Willmott, et al. Towards an argument interchange format. *The knowledge engineering review*, 21(4):293–316, 2006.

[29] Commission of the European Communities. A better functioning food supply chain in europe. Technical report, 2009.

[30] Marco Comuzzi, Jochem Vonk, and Paul Grefen. Measures and mechanisms for process monitoring in evolving business networks. *Data & Knowledge Engineering*, 71(1):1–28, 2012.

[31] Ştefan Conţiu and Adrian Groza. Improving remote sensing crop classification by argumentation-based conflict resolution in ensemble learning. *Expert Systems with Applications*, 64:269–286, 2016.

[32] Kuhn D. *The Skills of Argument.* Cambridge University Press, 1991.

[33] Wändi Bruine de Bruin and M Granger Morgan. Reflections on an interdisciplinary collaboration to inform public understanding of climate change, mitigation, and impacts. *Proceedings of the National Academy of Sciences*, page 201803726, 2019.

[34] Bruno de Moura Araujo, Eber Assis Schmitz, Alexandre Luis Correa, and Antonio Juarez Alencar. A method for validating the compliance of business processes to business rules. In *Proc. of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 145–149, New York, NY, USA, 2010. ACM.

[35] E. de Vries, K. Lund, and M. Baker. Computer-mediated epistemic dialogue: Explanation and argumentation as vehicles for understanding scientific notions. *Journal of the Learning Sciences*, 11(1):63–103, 2002.

[36] R. DeFries and J. Townshend. NDVI-derived land cover classifications at a global scale. *International Journal of Remote Sensing*, 15(17):3567–3586, 1994.

[37] Sara Delamont, Paul Atkinson, and Odette Parry. *Supervising the PhD: A Guide to Success.* ERIC, 1998.

[38] Paul E Dunne, Anthony Hunter, Peter McBurney, Simon Parsons, and Michael Wooldridge. Inconsistency tolerance in weighted argument systems.

In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 851–858. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[39] Paul E Dunne, Anthony Hunter, Peter McBurney, Simon Parsons, and Michael Wooldridge. Weighted argument systems: Basic definitions, algorithms, and complexity results. *Artificial Intelligence*, 175(2):457–486, 2011.

[40] Jan Eijck and Rineke Verbrugge. *Discourses on social software*. Amsterdam University Press, 2009.

[41] Dirk Fahland, Cedric Favre, Jana Koehler, Niels Lohmann, Hagen Volzer, and Karsten Wolf. Analysis on demand: Instantaneous soundness checking of industrial business process models. *Data & Knowledge Engineering*, 70(5):448–466, 2011.

[42] Melvin Fitting. Reasoning with justifications. In David Makinson, Jacek Malinowski, and Heinrich Wansing, editors, *Towards Mathematical Philosophy, Papers from the Studia Logica conference Trends in Logic IV*, volume 28 of *Trends in Logic*, chapter 6, pages 107–123. Springer, 2009. Published online November 2008.

[43] Christian Fleck. *Sociology in Austria since 1945*. Springer, 2015.

[44] Food and Agriculture Organisation of the United Nations World Health Organization. *Codex Alimentarus*. 1997.

[45] Giles M Foody. Thematic map comparison: Evaluating the statistical significance of differences in classification accuracy. *Photogrammetric engineering & remote sensing*, 70(5):627–633, 2004.

[46] Galileo Galilei. Dialogue concerning the two chief world systems, trans. stillman drake, 1953.

[47] Alejandro J García and Guillermo R Simari. Defeasible logic programming: An argumentative approach. *Theory and practice of logic programming*, 4(1+2):95–138, 2004.

[48] M. Georgiu and A. Groza. Ontology enrichment using semantic wikis and design patterns. *Romania*, 56(2):31–36, 2011.

[49] Sergio Alejandro Gomez, Anca Goron, Adrian Groza, and Ioan Alfred Letia. Assuring safety in air traffic control systems with argumentation and model checking. *Expert Systems with Applications*, 44:367 – 385, 2016.

[50] Sergio Alejandro Gomez, Adrian Groza, and Carlos Ivan Chesñevar. An argumentative approach to assessing safety in medical device software using defeasible logic programming. In *International Conference on Advancements of Medicine and Health Care through Technology; 5th–7th June 2014, Cluj-Napoca, Romania*, pages 167–172. Springer, 2014.

[51] T. F. Gordon. The pleadings game. *Artificial Intelligence and Law*, 2(4):239–292, 1993.

[52] T. F. Gordon, H. Prakken, and D. Walton. The Carneades model of argument and burden of proof. *Artificial Intelligence*, 171(10-15):875–896, 2007.

[53] A. Goron, A. Groza, S. A. Gomez, and I. A. Letia. Towards an argumentative approach for repair of hybrid logics models. In *Argumentation in Multi-Agent Systems*, 2014.

[54] Guido Governatori, Jörg Hoffmann, Shazia Wasim Sadiq, and Ingo Weber. Detecting regulatory compliance for business process models through semantic annotations. In *Business Process Management Workshop, 2008*, pages 5–17, 2008.

[55] A. Groza, I. Dragoste, I. Sincai, and I. Jimborean. An ontology selection and ranking system based on analytical hierarchy process. In *16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC2014), Timisoara, Romania, 22-25 September 2014.*, 2014.

[56] A. Groza, I. A. Letia, A. Goron, and S. Zaporojan. A formal approach for identifying assurance deficits in unmanned aerial vehicle software. In H. Selvaraj, D. Zydek, and G. Chmaj, editors, *In Proceedings of 23 International Conference on Systems Engineering, Las Vegas, USA*, volume 1089 of *Advances in Intelligent Systems & Computing Series*. Springer International Publishing, 2015.

[57] A. Groza and I.A. Letia. Agent-based systems for norm compliance in food supply chains. *Analele Universitatii de Vest din Timisoara*, 48(3), 2011.

[58] A. Groza and C. Man. Towards automatic norm compliance in construction domain. In *Applied Machine Intelligence and Informatics (SAMI), 2011 IEEE 9th International Symposium on*, pages 83–87. IEEE, 2011.

[59] A. Groza, A. Marginean, and V. Muresan. An ontology-based model for vehicular ad-hoc networks. In *18th IEEE International Conference on Intelligent Engineering Systems (INES2014), 3-5 July, Tihany, Hungary.* IEEE, 2014.

[60] A. Groza, B. Varga, and M. Vacca. A learning environment for building and evaluating ontologies: case study of 2013 Ontology Building Competition. In Ion Roceanu, editor, *E-learning and Software in Education (ELSE2014), Bucuresti, Romania*. "Carol I" National Defence University Publishing House, 2014.

[61] Adrian Groza. Modelling imprecise arguments in a weighted argument system. In Ioan Alfred Letia, editor, *5th International Conference on Intelligent Computer Communication and Processing*, pages 43–46, Cluj-Napoca, Romania, 2009.

[62] Adrian Groza. Data structuring for the ontological modelling of wind energy systems. In *4th Int. Conf. on Modelling and Development of Intelligent Systems (MDIS2015), Sibiu, Romania, 28 Oct. - 1 Nov. 2015*, 2015.

[63] Adrian Groza, Gabriel Barbur, and Blaga Bogdan. Ontology enrichment and evaluation using ontorich. *Automation Computers Applied Mathematics*, 2012.

[64] Adrian Groza and Lidia Corde. Information retrieval in falktales using natural language processing. In *Intelligent Computer Communication and Processing (ICCP), 2015 IEEE International Conference on*, pages 59–66. IEEE, 2015.

[65] Adrian Groza, Bogdan Iancu, and Anca Marginean. A multi-agent approach towards cooperative overtaking in vehicular networks. In Rajendra Akerkar, Nick Bassiliades, John Davies, and Vadim Ermolayev, editors, *WIMS*, page 48. ACM, 2014.

[66] Adrian Groza and Ioan Letia. Plausible description logics programs for stream reasoning. *Future Internet*, 4:865–881, 2012.

[67] Adrian Groza and Ioan Alfred Letia. Plausible description logic programs for stream reasoning. *Future Internet*, 4(4):865–881, 2012.

[68] Adrian Groza and Nicoleta Marc. Consistency checking of safety arguments in the goal structuring notation standard. In *IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP2014), Cluj-Napoca, Romania, 4-6 September 2014*, pages 59–66. IEEE, 2014.

[69] Adrian Groza and Madalina Mandy Nagy. Harmonization of conflicting medical opinions using argumentation protocols and textual entailment-a case study on parkinson disease. In *2016 IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 163–170. IEEE, 2016.

[70] Adrian Groza, Pinar Ozturk, Radu Razvan Slavescu, Anca Marginean, and Rajendra Prasath. Analysing climate change arguments using subjective logic. In *2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 37–44. IEEE, 2018.

[71] Adrian Groza and Oana Maria Popa. Mining arguments from cancer documents using natural language processing and ontologies. In *2016 IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 77–84. IEEE, 2016.

[72] Adrian Groza and Mechno Raluca. Argumentation based ontology maintenance. In *4th Int. Conf. on Modelling and Development of Intelligent Systems (MDIS2015), Sibiu, Romania, 29 Oct. - 1 Nov. 2011*, 2011.

[73] Adrian Groza and Roxana Szabo. Enacting textual entailment and ontologies for automated essay grading in chemical domain. In *2015 16th IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 221–226. IEEE, 2015.

[74] Adrian Groza and Iulia Ungur. Improving conflict resolution in version spaces for precision agriculture. *International Journal of Agricultural Science*, 3, 2018.

[75] Serban Groza and Adrian Groza. Prograph: towards enacting bipartite graphs for abstract argumentation frameworks. In Matthias Thimm and Serena Villata, editors, *System Descriptions of the First International Competition on Computational Models of Argumentation (ICCMA'15)*, Advances in Intelligent Systems & Computing Series. arXiv preprint arXiv:1510.05373, 2015.

[76] Cally Guerin, Heather Kerr, and Ian Green. Supervision pedagogies: narratives from the field. *Teaching in Higher Education*, 20(1):107–118, 2015.

[77] Rolf Haenni. Probabilistic argumentation. *Journal of Applied Logic*, 7(2):155 – 176, 2009. Special issue: Combining Probability and Logic.

[78] Christine Halse. Becoming a supervisor: the impact of doctoral supervision on supervisors' learning. *Studies in higher education*, 36(5):557–570, 2011.

[79] Lawrence C Hamilton. Did the arctic ice recover? demographics of true and false climate facts. *Weather, Climate, and Society*, 4(4):236–249, 2012.

[80] Zhiyong Hao, Bin Liu, Junfeng Wu, and Jinhao Yao. Exploiting ontological reasoning in argumentation based multi-agent collaborative classification. In *Intelligent Information and Database Systems*, pages 23–33. Springer, 2015.

[81] C. G. Hempel and P. Oppenheim. Studies in the logic of explanation. *Philosophy of Science*, 15(2):135–175, 1948.

[82] Douglas R Hofstadter et al. *Gödel, Escher, Bach: an eternal golden braid:[a metaphorical fugue on minds and machines in the spirit of Lewis Carroll]*. Penguin Books New York, 1980.

[83] Oxana Hotea and Adrian Groza. Reasoning on semantic sensor streams for smart city. In *International Conference on Intelligent Information Systems, Chisinau, Republic of Moldova, August 20-23, 2013, ISBN 978-9975-4237-1-7,*, pages 219–222, 2013.

[84] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multi-class support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.

[85] Anthony Hunter. Real arguments are approximate arguments. In *AAAI*, volume 7, pages 66–71, 2007.

[86] Sergiu Indrie and Adrian Groza. Enacting argumentative web in semantic wikipedia. In Remus Brad, editor, *9th RoEduNet International Conference*, pages 163–168, Sibiu, Romania, 2010.

[87] Sergiu Indrie and Adrian Groza. Towards social argumentative machines. In Ioan Alfred Letia, editor, *6th International Conference on Intelligent Computer Communication and Processing*, pages 99–102, Cluj-Napoca, Romania,

2010.

[88] M. Janvid. Knowledge versus understanding: The cost of avoiding Gettier. *Acta Analytica*, 27:183–197, 2012.

[89] Ioana Jimborean and Adrian Groza. Ranking ontologies in the ontology building competition boc 2014. In *IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP2014), Cluj-Napoca, Romania, 4-6 September 2014*, pages 75–82. IEEE, 2014.

[90] Clessa J.J. *Math and Logic Puzzles for PC Enthusiasts*. Courier Corporation, 1996.

[91] Audun Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(03):279–311, 2001.

[92] Dimitris Karagiannis. A business process-based modelling extension for regulatory compliance. In *Multikonferenz Wirtschaftsinformatik, 2008*, 2008.

[93] Joel Katzav and Chris A Reed. On argumentation schemes and the natural classification of arguments. *Argumentation*, 18(2):239–259, 2004.

[94] Marwane El Kharbili, Ana Karla A. de Medeiros, Sebastian Stein, and Wil M. P. van der Aalst. Business process compliance checking: Current state and future challenges. In *MobIS, 2008*, pages 107–113, 2008.

[95] Andrei P Kirilenko and Svetlana O Stepchenkova. Public microblogging on climate change: One year of twitter worldwide. *Global Environmental Change*, 26:171–182, 2014.

[96] Boris A Kordemsky. *The Moscow puzzles: 359 mathematical recreations*. Courier Corporation, 1992.

[97] Ludmila I Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2004.

[98] Boris Lauser, Gudrun Johannsen, Caterina Caracciolo, Willem Robert van Hage, Johannes Keizer, and Philipp Mayr. Comparing human and automatic thesaurus mapping approaches in the agricultural domain. *Universitätsverlag Göttingen*, page 43, 2008.

[99] Matthew Lease. On quality control and machine learning in crowdsourcing. *Human Computation*, 11:11, 2011.

[100] Ioan Alfred Leţia and Adrian Groza. Arguing with justifications between collaborating agents,. In P. McBurney et al. (eds), editor, *ARGMAS, 2012)*, volume 7543 of *LNAI*, pages 102–116. Springer, 2012.

[101] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[102] John Lee, Finbar Dineen, and Jean McKendree. Supporting student discussions: it isn't just talk. *Education and Information Technologies*, 3(3-4):217–229, 1998.

[103] I. A. Letia and A. Groza. Interleaved argumentation and explanation in dialog. In *Computational Models of Natural Argument*, pages 44–52, 2012.

[104] Ioan Alfred Letia and Adrian Groza. Contextual extension with concept maps in the argument interchange format. In Iyad Rahwan and Pavlos Moraitis, editors, *ArgMAS*, volume 5384 of *Lecture Notes in Computer Science*, pages 72–89. Springer, 2008.

[105] Ioan Alfred Letia and Adrian Groza. Modelling imprecise arguments in description logic. *Advances in Electrical and Computer Engineering*, 9(3):94–99, 2009.

[106] Ioan Alfred Letia and Adrian Groza. Argumentative support for structured HACCP plans. *Advances in Electrical and Computer Engineering*, 10(2):115–120, 2010.

[107] Ioan Alfred Letia and Adrian Groza. Argumentative support for structured HACCP plans. In *International Conference on Development and Application Systems*, Suceava, Romania, 2010.

[108] Ioan Alfred Letia and Adrian Groza. Developing hazard ontology for supporting haccp systems in food supply chains. In *8th IEEE International Symposium on Intelligent Systems and Informatics*, pages 57–62, Subotica, Serbia, 2010.

[109] Ioan Alfred Letia and Adrian Groza. Towards pragmatic argumentative agents within a fuzzy description logic framework. In *Seventh International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2010)*, Toronto, Canada, 2010.

[110] Ioan Alfred Letia and Adrian Groza. Towards pragmatic argumentative agents within a fuzzy description logic framework. *Argumentation in Multi-Agent Systems*, pages 209–227, 2011.

[111] Ioan Alfred Letia and Adrian Groza. *Arguing with Justifications between Collaborating Agents*, pages 102–116. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[112] Ioan Alfred Letia and Adrian Groza. Description plausible logic programs for stream reasoning. In Joaquim Filipe and Ana L. N. Fred, editors, *ICAART (1)*, pages 560–566. SciTePress, 2012.

[113] Ioan Alfred Letia and Adrian Groza. Interleaved argumentation and explanation in dialog. In *the! 12th! workshop! on!! Computational! Models! of! Natural! Argument!*, page 44, 2012.

[114] Ioan Alfred Letia and Adrian Groza. Justificatory argumentation for commitment agents. *Argumentation in Multi-Agent Systems*, 2012.

[115] Ioan Alfred Letia and Adrian Groza. Compliance checking of integrated business processes. *Data Knowl. Eng.*, 87:1–18, 2013.

[116] Ioan Alfred Letia and Adrian Groza. Compliance checking of integrated

business processes. *Data & Knowledge Engineering*, 87(0):1 – 18, 2013.
[117] Ioan Alfred Letia, Adrian Groza, and Radu Balaj. Argumentative agents for justifying decisions in audit. In *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on*, pages 71–78. IEEE, 2011.
[118] Florin Lipan and Adrian Groza. Mining traffic patterns from public transportation GPS data. In Ioan Alfred Letia, editor, *6th International Conference on Intelligent Computer Communication and Processing*, pages 123–126, Cluj-Napoca, Romania, 2010.
[119] Ruopeng Lu, Shazia Wasim Sadiq, and Guido Governatori. Measurement of compliance distance in business processes. *IS Management*, 25(4):344–355, 2008.
[120] C. Lumer. The epistemological theory of argument: How and why? *Informal Logic*, 25:213–242, 2005.
[121] Linh Thao Ly, Stefanie Rinderle, and Peter Dadam. Integration and verification of semantic constraints in adaptive process management systems. *Data & Knowledge Engineering*, 64(1):3–23, 2008.
[122] Bernardo Magnini, Roberto Zanoli, Ido Dagan, Kathrin Eichler, Günter Neumann, Tae-Gil Noh, Sebastian Pado, Asher Stern, and Omer Levy. The excitement open platform for textual inferences. In *ACL (System Demonstrations)*, pages 43–48, 2014.
[123] Michael J Maher. Human and unhuman commonsense reasoning. In *International Conference on Logic for Programming Artificial Intelligence and Reasoning*, pages 16–29. Springer, 2010.
[124] Catherine Manathunga*. The development of research supervision:âĂIJturning the light on a private spaceâĂİ. *International Journal for Academic Development*, 10(1):17–30, 2005.
[125] A Marginean, A Groza, R.R. Slavescu, and IA Letia. Romanian2sparql: A grammatical framework approach for querying linked data in romanian. In *Development and Application Systems (DAS), 2014 International Conference on*, pages 204–209, May 2014.
[126] G. R. Mayes. Resisting explanation. *Argumentation*, 14:361–380, 2000.
[127] Gregory Randolph Mayes. Argument explanation complementarity and the structure of informal reasoning. *Informal Logic*, 30(1):92–111, 2010.
[128] William McCune. Mace4 reference manual and guide. *arXiv preprint cs/0310055*, 2003.
[129] Michael Mendler and Stephan Scheele. Towards constructive DL for abstraction and refinement. *Journal of Automated Reasoning*, 44(3):207–243, 2010.
[130] S. Modgil, F. Toni, F. Bex, I. Bratko, C. Chesnevar, W. Dvorak, M. Falappa,

X. Fan, S. Gaggl, A. Garcia, M. Gonzalez, T. Gordon, J. Leite, M. Mozina, C. Reed, G. Simari, S. Szeider, P. Torroni, and S. Woltran. The added value of argumentation. In S. Ossowski, editor, *Agreement Technologies*, volume 8, pages 357–403. Springer, 2013.

[131] Sanjay Modgil. Hierarchical argumentation. In Michael Fisher, Wiebe van der Hoek, Boris Konev, and Alexei Lisitsa, editors, *JELIA*, volume 4160 of *Lecture Notes in Computer Science*. Springer, 2006.

[132] Sanjay Modgil and Trevor J. M. Bench-Capon. Metalevel argumentation. *Journal of Logic and Computation*, 2010.

[133] Saif M Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. A dataset for detecting stance in tweets. In *Proceedings of 10th edition of the the Language Resources and Evaluation Conference (LREC), Portoroz, Slovenia*, 2016.

[134] Don Monroe. Ai, explain yourself. *Communications of the ACM*, 61(11):11–13, 2018.

[135] Marco Montali. *Specification and Verification of Declarative Open Interaction Models*. Springer, 2010.

[136] Erik T Mueller. *Commonsense reasoning: an event calculus based approach*. Morgan Kaufmann, 2014.

[137] Kioumars Namiri and Nenad Stojanovic. Towards a formal framework for business process compliance. In *Multikonferenz Wirtschaftsinformatik 2008*, pages 344–355. GTO-Verlag, 2008.

[138] USDA NASS. Field crops: Usual planting and harvesting dates. *USDA National Agricultural Statistics Service, Agriculural Handbook*, (628), 2010.

[139] Sebastian Padó, Tae-Gil Noh, Asher Stern, Rui Wang, and Roberto Zanoli. Design and realization of a modular architecture for textual entailment. *Natural Language Engineering*, 21(02):167–200, 2015.

[140] Fabio Paglieri and Cristiano CastelFranchi. Why arguing? towards a cost-benefit analysis of argumentation. *Argument and Computation*, 1(1):71–91, 2010.

[141] V. Pankratius, J. Li, M. Gowanlock, D. M. Blair, C. Rude, T. Herring, F. Lind, P. J. Erickson, and C. Lonsdale. Computer-aided discovery: Toward scientific insight generation with machine support. *IEEE Intelligent Systems*, 31(4):3–10, July 2016.

[142] R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45, Third 2006.

[143] Shannon Pollard and Robert C Duvall. Everything i needed to know about teaching i learned in kindergarten: bringing elementary education techniques to undergraduate computer science classes. In *ACM SIGCSE Bulletin*, volume 38, pages 224–228. ACM, 2006.

[144] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2):93–124, 2010.

[145] Henry Prakken. A study of accrual of arguments, with applications to evidential reasoning. In *Proceedings of the 10th international conference on Artificial intelligence and law*, pages 85–94. ACM, 2005.

[146] Iyad Rahwan and Bita Banihashemi. Arguments in owl: A progress report. *COMMA*, 172:297–310, 2008.

[147] Iyad Rahwan, Fouad Zablith, and Chris Reed. Laying the foundations for a world wide argument web. *Artif. Intell.*, 171(10-15):897–921, 2007.

[148] Sergiu Redeca and Adrian Groza. Designing agents for the stratego game. In *2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 97–104. IEEE, 2018.

[149] Chris Reed and Glenn Rowe. Araucaria: Software for argument analysis, diagramming and representation. *International Journal on Artificial Intelligence Tools*, 13(04):961–979, 2004.

[150] Bryan Renne. Evidence elimination in multi-agent justification logic. In *Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge*, TARK '09, pages 227–236, New York, NY, USA, 2009. ACM.

[151] Jason Rosenhouse. *The Monty Hall problem: the remarkable story of Math's most contentious brain teaser*. Oxford University Press, 2009.

[152] M. Ventura A. Graesser S. Craig, B. Gholson. Overhearing dialogues and monologues in virtual tutoring sessions: Effects on questioning and vicarious learning. *International Journal of Artificial Intelligence in Education*, 11:242–253, 2000.

[153] O. Scheuer, F. Loll, N. Pinkwart, and B. M. McLaren. Computer-supported argumentation: A review of the state of the art. *Computer-Supported Collaborative Learning*, 5(1):43–102, 2010.

[154] Rudy Setiono and Huan Liu. Neurolinear: From neural networks to oblique decision rules. *Neurocomputing*, 17(1):1–24, 1997.

[155] Yoav Shoham. Why knowledge representation matters. *Communications of the ACM*, 59(1):47–49, 2015.

[156] Raymond M Smullyan. *The Tao is silent*. Harper & Row New York, 1977.

[157] Asher Stern and Ido Dagan. The BIUTEE research platform for transformation-based textual entailment recognition. *LiLT (Linguistic Issues in Language Technology)*, 9, 2014.

[158] Asher Stern, Roni Stern, Ido Dagan, and Ariel Felner. Efficient search for transformation-based inference. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 283–291. Association for Computational Linguistics, 2012.

[159] Daniel Suciu and Adrian Groza. Interleaving ontology-based reasoning and

natural language processing for character identification in folktales. In *IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP2014), Cluj-Napoca, Romania, 4-6 September 2014*, pages 67–74. IEEE, 2014.

[160] Roxana Szabo and Adrian Groza. Analysing debates on climate change with textual entailment and ontologies. In *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 39–46. IEEE, 2017.

[161] Daniel Toniuc and Adrian Groza. Climebot: An argumentative agent for climate change. In *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 63–70. IEEE, 2017.

[162] Stephen E Toulmin. *The uses of argument.* Cambridge university press, 2003.

[163] David S Touretzky. *COMMON LISP: A Gentle Introduction to Symbolic Computation.* Courier Corporation, 2013.

[164] United States Department of Agriculture. Census of agriculture, New Madrid County, Missouri. 2012.

[165] Wil MP van der Aalst, Martin Bichler, and Armin Heinzl. Responsible data science, 2017.

[166] Thomas L. van der Weide, Frank Dignum, John-Jules Ch. Meyer, Henry Prakken, and Gerard Vreeswijk. Multi-criteria argument selection in persuasion dialogues. In *AAMAS*, pages 921–928, 2011.

[167] B. Varga and A. Groza. Integrating dbpedia and sentiwordnet for a tourism recommender system. In *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on*, pages 133–136. IEEE, 2011.

[168] A. Ramirez Vela and J. Martin Fernandez. Barriers for the developing and implementation of HACCP plans: results from a spanish regional survey. *Food Control*, 14(5):333–337, 2003.

[169] Gabriela Visinari and Adrian Groza. Semantic-based monitoring of e-contracts. In *10th National Conference on Human - Computer Interaction ROCHI 2013*, pages 161–164, 2013.

[170] D. Walton. A dialogue system specification for explanation. *Synthese*, 182(3):349–374, 2011.

[171] Douglas Walton. *Fundamentals of critical argumentation.* Cambridge University Press, 2005.

[172] Douglas Walton and David M. Godden. Redefining knowledge in a way suitable for argumentation theory. In H.V. Hansen, editor, *Dissensus and the Search for Common Ground*, pages 1–13, 2007.

[173] Douglas Walton, Chris Reed, and Fabrizio Macagno. *Argumentation*

Schemes. Cambridge University Press, 2008.

[174] Maya Wardeh, Frans Coenen, and Trevor Bench Capon. PISA: A framework for multiagent classification using argumentation. *Data & Knowledge Engineering*, 75:34–57, 2012.

[175] Matt Webster, Michael Fisher, Neil Cameron, and Mike Jump. Formal methods for the certification of autonomous unmanned aircraft systems. In *Computer Safety, Reliability, and Security*, pages 228–242. Springer, 2011.

[176] Matt Webster, Michael Fisher, Neil Cameron, and Mike Jump. Model checking and the certification of autonomous unmanned aircraft systems. Technical Report ULCS-11-001, Department of Computer Science, University of Liverpool, Liverpool, United Kingdom, 2011.

[177] Kathleen D Weiss, Christina R Vargas, Olivia A Ho, Danielle J Chuang, Jonathan Weiss, and Bernard T Lee. Readability analysis of online resources related to lung cancer. *Journal of Surgical Research*, 206(1):90–97, 2016.

[178] Michael Wooldridge, Peter McBurney, and Simon Parsons. On the meta-logic of arguments. In *AAMAS*, pages 560–567, 2005.

[179] L. Wright. Reasoning and explaining. *Argumentation*, 16:33–46, 2002.

[180] Tatiana Yavorskaya. Interacting explicit evidence systems. *Theory of Computer Systems*, 43:272–293, 2008.

[181] Huiling Zhang, Md Abdul Alim, Xiang Li, My T. Thai, and Hien T. Nguyen. Misinformation in online social networks: Detect them all with a limited budget. *ACM Trans. Inf. Syst.*, 34(3):18:1–18:24, April 2016.

[182] Yan Zhang and Yulin Ding. CTL model update for system modifications. *J. Artif. Intell. Res.(JAIR)*, 31:113–155, 2008.