Automated Generation of Erotetic Search Scenarios: Classification, Optimisation and Knowledge Extraction

Szymon Chlebowski Maciej Komosinski Adam Kups

Sz. Chlebowski and A. Kups: Adam Mickiewicz University, Institute of Psychology, Chair of Logic and Cognitive Science, Szamarzewskiego 89a, 60-568 Poznan, Poland, {szymon.chlebowski, adamku}@amu.edu.pl. M. Komosinski: Institute of Computing Science, Piotrowo 2, 60-965 Poznan, Poland, maciej.komosinski@cs.put.poznan.pl

Abstract

This paper concerns automated generation and processing of *erotetic search scenarios* (ESSs). ESSs are formal constructs characterized in Inferential Erotetic Logic that enable finding possible answers to a posed question by decomposing it into auxiliary questions. The first part of this work describes a formal account on ESSs. The formal approach is then applied to automatically generate ESSs, and the resulting scenarios are evaluated according to a number of criteria. These criteria are subjected to discordance analysis that reveals their mutual relationships. Finally, knowledge concerning relationships between different values of evaluation criteria is extracted by applying Apriori – an association rules mining algorithm. The proposed approach of integration of formal erotetic logic with computational tools provides an extensive insight into the former and helps with the development of efficient ESSs.

Contents

1 Introduction **2**

2 Inferential Erotetic Logic and Erotetic Search Scenarios **2**

3 Embedding and Contraction **5**
3.1 Embedding 5
3.2 Contraction 8

4 Redundancy elimination **9**
4.1 Co-occurrence of two queries 10
4.2 Co-occurrence of a query or an auxiliary question and an auxiliary question 12
4.3 Co-occurrence of a query and an answer to this query 13

5 Automated generation and analyses of ESSs **13**
5.1 Automatic generation of ESSs 14
5.2 Evaluation criteria for ESSs 15
5.3 Multi-criteria evaluation 18
5.4 Multi-criteria dominance relation 19
5.5 Multi-criteria discordance analysis 21
5.6 Mining relationships between criteria 22

6 Conclusions **23**

This paper has been published in *ACM Transactions on Computational Logic* 18 (2), 2017.
http://dx.doi.org/10.1145/3056537

1 Introduction

One of the interesting and pending issues on the verge of contemporary logic and informatics is posing and answering questions in different disciplines such as problem solving, natural language processing or using digital databases. Processing of questions can be formally described using the logic of questions. There are various sophisticated formal frameworks which are used to model questions and answers, such as [4, 22, 17] among many others (for an overview, see [16] and [36, 40]). Currently the most popular and well-developed approaches are the Interrogative Model of Inquiry which uses game-theoretical tools [18, 19, 2] and the Inquisitive Semantics which introduces concepts of *informative* and *inquisitive* content of a sentence defined in the framework of *possible-world semantics* [7, 6, 8]. There are also various approaches to apply tools of *dynamic* and/or *epistemic* logic in erotetic logic [31, 35, 14]. Questions and their logic plays an important role also in dialogue theory [11, 12].

This work introduces tools developed on the grounds of Wiśniewski's Inferential Erotetic Logic (IEL) [40] – a logic in which different kinds of inferences involving questions are analysed in remarkable detail, which distinguishes it from concurrent approaches. IEL has much to offer for researchers interested in fields such as proof theory (for example an application of IEL to proof theory of classical logic [39, 41, 27], modal logics [24, 25], paraconsistent logics [5]), artificial intelligence [28, 30], and philosophy of science and theory of explanation [23, 37, 34]. In particular, IEL introduces the concept of *erotetic search scenarios* (e-scenarios, ESSs) – diagrams which represent possible strategies of solving a given problem (which are in fact strategies of obtaining answers to a posed question).

The computational tools for e-scenarios described in this paper are able to perform quantitative analyses of question processing in formal systems. Such tools are key components of a platform that will provide e-scenarios for use in various human–computer interfaces using natural language processing.

The primary contribution of this work (Sect. 4) is a formal introduction of new operations that eliminate redundancies and contradictions from ESSs. These operations are implemented alongside the embedding and contraction operations, which are used for automatic generation and optimisation of e-scenarios. The tools for generation of ESSs produce large amounts of e-scenarios and their evaluations that allow for analyses of these formal constructs. Another major contribution of this work (Sect. 5) is the demonstration of how qualitative and quantitative data that result from classification of e-scenarios can be used to find interesting patterns and relationships between different types of e-scenarios. This is the first work to show that erotetic search scenarios can be automatically obtained, evaluated, optimised, and therefore effectively researched. Introducing useful tools designed for the development of IEL is the next step of introducing computational methodology to various fields of philosophical and mathematical logic [21, 20].

2 Inferential Erotetic Logic and Erotetic Search Scenarios

Wiśniewski's Inferential Erotetic Logic [40] (IEL for short) is a logical framework for modelling the types of reasoning in which questions play an important role. In IEL, questions are formalized by adding to the language of a given logic L[1] a new set of symbols (?, {, }), and by defining a new kind of well-formed expressions – i.e., *erotetic formulas*. The problem of the relation between questions and answers is solved by means of Hamblin's postulate: *Knowing what counts as an answer is equivalent to knowing the question* [15, 40]. This enables one to represent a question Q as a list of possible answers to Q. Thus *erotetic well–formed formulas* (e-formulas or questions for short) are of the form:

$$Q = ?\{A_1, A_2, \ldots, A_n\}$$

where $n > 1$ and $A_1, \ldots A_n$ are (intuitively) *direct answers*[2] to Q. We require that they are pairwise syntactically distinct declarative formulas of a language of a given logic L. Note that questions $Q_1 = ?\{A, \neg A\}$

[1] For the sake of simplicity, in this paper we only consider Classical Propositional Logic (CPL).

[2] The characterization of direct answers differs from one theory of questions to another. What seems to be important is that direct answers "*are directly and precisely responsive to the question, giving neither more nor less information than what is called for*" [3, p. 124].

and $Q_2 = ?\{\neg A, A\}$ are not the same (as they are different expressions of an object level language), nevertheless the set of direct answers to question Q_1 ($\mathsf{d}Q_1$ for short) is the same as the set of direct answers to Q_2, since $\mathsf{d}Q_1 = \mathsf{d}Q_2 = \{A, \neg A\}$.

Let v be a classical valuation. It is said that question Q is *sound under* v iff there exists $A \in \mathsf{d}(Q)$ such that $v(A) = 1$. A question Q is *sound* iff for each classical valuation v there exists $A \in \mathsf{d}(Q)$ such that $v(A) = 1$. A question which is not sound is called *risky.*

Let us look at some examples:

1. A question *Does Adam like black metal?* can be formalized by the following e-formula: $?\{p, \neg p\}$, where p means *Adam likes black metal.* Naturally, this question is sound.

2. A question *What kind of music does Simon like: black metal or noise?* can be formalized by different e-formulas depending of our understanding of this question (is it possible for him to like both or neither?): (i) $?\{p, q\}$, (ii) $?\{p, q, p \wedge q\}$, (iii) $?\{p, q, \neg(p \vee q)\}$, (iv) $?\{p \wedge q, \neg p \wedge q, p \wedge \neg q, \neg p \wedge \neg q\}$ (where p means *Simon likes black metal* and q means that *Simon likes noise*). Questions (iii) and (iv) are sound, while (i) and (ii) are risky.

Perhaps the most useful concept of IEL is that of *erotetic implication.* It is said that a question Q implies a question Q^* on the basis of a possibly empty set of declarative formulas (d-wffs for short) X if and only if (1) each direct answer to the question Q entails, together with X, a set of all direct answers to the question Q^* (*transmission of soundness*), and (2) for each direct answer A to the question Q^* there exists a proper subset Y of the set of all direct answers to the question Q such that an answer A along with X entails Y (*open-minded cognitive usefulness*). The underlying intuition is that answering an *entailed question* brings us closer to get an answer to the *implying question* or, to put it in another words, the set of possibilities offered by the implied question *narrows down* the set of possibilities offered by the implying question.

To define what it means for a set of formulas to be implied by a set of formulas we use the concept of *multiple-conclusion entailment* [32]. We will say that $X \Vdash Y$ if and only if for each classical valuation v, $v(A) = 0$ for some $A \in X$, or $v(B) = 1$ for some $B \in Y$.

Definition 2.1 (Erotetic implication, e-implication)**.** *Let Q, Q^* be questions and X be the set of declarative well-formed formulas. A question Q e-implies a question Q^* on the basis of a possibly empty set of declarative formulas X ($\mathbf{Im}(Q, X, Q^*)$) iff*

1. *for each $A \in \mathsf{d}Q$: $X \cup \{A\} \Vdash \mathsf{d}Q_1$, and*

2. *for each $B \in \mathsf{d}Q_1$ there exists a non–empty proper subset Y of $\mathsf{d}Q$ such that $X \cup \{B\} \Vdash Y$.*

Erotetic Decomposition Principle (*EDP*) states that a suitable way to resolve a principal question is to transform it, by means of erotetic implication, into auxiliary questions dependent on the principal question or one of the previously stated auxiliary questions and (optionally) answers to the previous auxiliary questions. The resolution of auxiliary questions leads to the resolution of the principal question [40].

The formalisation of the EDP is the concept of the Erotetic Search Scenario based on IEL. Intuitively speaking, an e-scenario (ESS) might be seen as a certain strategy prepared to find the answer to a posed question depending on the available knowledge. In order to solve this problem we need to follow the path of the scenario from the root and consult our knowledge at the branching points to choose which direction to follow further – until we arrive at a leaf which is the answer to our question according to the available knowledge.

Let us consider an example presented in Fig. 1, where we ask whether it is the case that $(p \wedge q)$ or not. The ESS is created according to the EDP and conforms to the erotetic implication requirements, which guarantees soundness of each step of the answer search procedure. Directly because of those requirements, the second (auxiliary) question – $?\{p \wedge q, \neg p \wedge q, p \wedge \neg q, \neg p \wedge \neg q\}$ – is included in the scenario. While posing the question $?\{p, \neg p\}$ right after the main question might look straightforward and intuitive, it does not conform to the second requirement of the erotetic implication. Only after inclusion of the auxiliary question (which is erotetically implied by the main question) we can derive the question $?\{p, \neg p\}$ (which is erotetically implied by the auxiliary question). Posing an auxiliary question may be seen as a kind of an attempt to clarify the state of affairs implied by the previous information and, in the described example, it demonstrates that erotetic implication is not transitive.

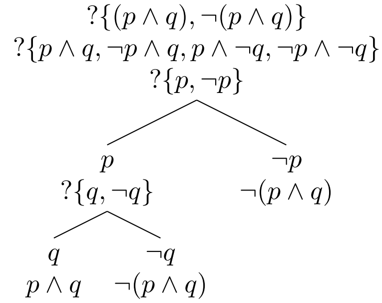


Figure 1: A sample e-scenario.

An e-scenario may be described as a labelled tree with a principal question at its root and possibly some d-wffs (representing background knowledge; none of them can be an answer to the principal question) labelling consecutive nodes. Then, each consecutive node is labelled by a formula that is either: (1) a d-wff derived from the previous d-wffs or a direct answer to the preceding auxiliary question; (2) an e-formula which is erotetically implied by the previous questions (possibly on the basis of previous d-wffs). Every node labelled by a d-wff can be followed by at most one child-node. Each node labelled by an e-formula has either one child node if it is labelled by an e-formula, or several child nodes labelled by all the direct answers to the question represented by that node – a question represented by such an e-formula is called a query. Each leaf of such tree has to be labelled by a direct answer to the principal question. A properly built e-scenario has to contain at least one query.

From the meta-logical point of view it is plausible to define e-scenarios as certain families of e-derivations – i.e., as sets of sequences of formulas. These two approaches are equivalent [26].

Definition 2.2 (E-derivation)**.** *A finite sequence* $\mathbf{s} = \mathbf{s}_1, \ldots, \mathbf{s}_n$ *of wffs is an* erotetic derivation *(e-derivation) of a direct answer* A *to the question* Q *on the basis of a set of d-wffs* X *iff* $\mathbf{s}_1 = Q$, $\mathbf{s}_n = A$ *and the following conditions are satisfied:*

1. *for each question* $\mathbf{s}_k$ *of* $\mathbf{s}$ *such that* $k > 1$*:*
   (a) $\mathsf{d}\mathbf{s}_k \neq \mathsf{d}Q$,
   (b) $\mathbf{s}_k$ *is implied by a certain question* $\mathbf{s}_j$ *which precedes* $\mathbf{s}_k$ *in* $\mathbf{s}$ *on the basis of the empty set, or on the basis of a non-empty set of d-wffs such that each element of this set precedes* $\mathbf{s}_k$ *in* $\mathbf{s}$*, and*
   (c) $\mathbf{s}_{k+1}$ *is either a direct answer to* $\mathbf{s}_k$ *or a question.*
2. *For each d-wff* $\mathbf{s}_i$ *of* $\mathbf{s}$*:*
   (a) $\mathbf{s}_i \in X$*, or*
   (b) $\mathbf{s}_i$ *is a direct answer to* $\mathbf{s}_{i-1}$*, where* $\mathbf{s}_{i-1} \neq Q$*, or*
   (c) $\mathbf{s}_i$ *is entailed by a certain non-empty set of d-wffs such that each element of this set precedes* $\mathbf{s}_i$ *in* $\mathbf{s}$.

A *query* of a derivation is a question of that e-derivation, which is immediately followed by an answer. Each question in an e-derivation, which is not the initial question and not a query, is called an *auxiliary question.*

Definition 2.3 (Erotetic search scenario)**.** *A finite family* Σ *of sequences of wffs is an* erotetic search scenario *(e-scenario) for a question* Q *relative to a set of d-wffs* X *iff each element of* Σ *is an e-derivation of a direct answer to* Q *on the basis of* X *and the following conditions hold:*

1. $\mathsf{d}Q \cap X = \emptyset$*,*

2. Σ *contains at least two elements,*

3. for each element $\mathbf{s} = \mathbf{s}_1, \ldots, \mathbf{s}_n$ *of* Σ *for each index* k*, where* $1 \leq k < n$*:*

   (a) if $\mathbf{s}_k$ *is a question and* $\mathbf{s}_{k+1}$ *is a direct answer to* $\mathbf{s}_k$*, then for each direct answer* B *to* $\mathbf{s}_k$*: the family* Σ *contains a certain e-derivation* $\mathbf{s}^* = \mathbf{s}^*_1, \ldots, \mathbf{s}^*_m$ *such that* $\mathbf{s}_j = \mathbf{s}^*_j$ *for* $j = 1, \ldots, k$*, and* $\mathbf{s}^*_{k+1} = B$*,*

   (b) if $\mathbf{s}_k$ *is a d-wff, or* $\mathbf{s}_k$ *is a question and if* $\mathbf{s}_{k+1}$ *is not a direct answer to* $\mathbf{s}_k$*, then for each e-derivation* $\mathbf{s}^* = \mathbf{s}^*_1, \ldots, \mathbf{s}^*_m$ *in* Σ *such that* $\mathbf{s}_j = \mathbf{s}^*_j$ *for* $j = 1, \ldots, k$ *we have* $\mathbf{s}_{k+1} = \mathbf{s}^*_{k+1}$*.*

Erotetic search scenarios have the following remarkable feature called the *Golden Path* property [40, p.116]. If Σ is an e-scenario for a question Q relative to a set of declarative formulas X, Q is sound (there is at least one true direct answer to Q) and formulas in X are true, then there exists a path in Σ (e-derivation) such that each question on that path is sound, each declarative formula on this path is true, and this path ends with a true direct answer to the initial question Q. In other words: if one poses a reasonable question Q, then an e-scenario for Q yields a strategy for obtaining a true answer to Q.

Definition 2.4. *An e-scenario* Σ *for a question* Q *relative to a set of declarative formulas* X *is said to be* complete *if and only if each direct answer to the question* Q *is the last term of a path of* Σ*.*

A sample e-scenario that is complete was shown in Fig. 1. In the following sections we make use of *standard e-scenarios*, which are pure (the set of initial declarative premises is empty) and have the initial question in one of the two forms: $?\{\neg A, \neg\neg A\}$ or $?\{A \otimes B, \neg(A \otimes B)\}$ (where $\otimes \in \{\wedge, \vee, \rightarrow\}$). In standard e-scenarios we aim at giving the answer to the yes–no question about the truth of a compound formula. An e-scenario in Fig. 1 is a standard e-scenario for conjunction.

3 Embedding and Contraction

There are two canonical operations which may be performed on e-scenarios: *embedding* of one e-scenario into another and *contraction* of an e-scenario by a selected answer to a query. Embedding can be performed when there exists an e-scenario **B** for the question which is a query of the e-scenario **A**. Contraction of an e-scenario by the selected answer to a query amounts to removal of all the paths started by the query except the path containing that answer, and removing the considered query. Both operations have to conform to several restrictions and usually demand more or less advanced syntactic transformations in order to produce valid e-scenarios.

Those operations when combined and automatised may be used for generation of new e-scenarios from already existing base scenarios by the Embedding Theorem and the Contraction Theorem. However, while the base scenarios may be well-optimised, the resulting scenarios can have some undesired properties – like redundancy and inconsistent formulas on some paths. Because of that, additional e-scenario repairing operations have to be introduced.

3.1 Embedding

We start with the embedding procedure announced in [38] and defined in detail in [40]. Since concepts closely connected to the notion of embedding are often used in this section, we replicate here this definition and the Embedding Theorem. Let us first introduce some auxiliary notions. $\Sigma_{[\mathbf{s},\mathbf{s}_k]}$ is the set of all paths of Σ which, firstly, have the query $\mathbf{s}_k$ as the k-th term and, secondly, agree with path $\mathbf{s}$ as to previous items ($\widehat{\Sigma}_{[\mathbf{s},\mathbf{s}_k]}$ is the complement of this set). $\Sigma_{[\mathbf{s},\mathbf{s}_k,B]}$ is the set of all paths of the set $\Sigma_{[\mathbf{s},\mathbf{s}_k]}$ which have a formula B as their $(k+1)$st terms. Naturally, $\Sigma_{[\mathbf{s},\mathbf{s}_k,B]} \subset \Sigma_{[\mathbf{s},\mathbf{s}_k]}$.

Fact 3.1. *Each path* $\mathbf{t} \in \Sigma_{[\mathbf{s},\mathbf{s}_k,B]}$ *can be represented as follows:*

$$\mathbf{t} = \gamma_{[\mathbf{t}]} \ ' \ \mathbf{t}_j \ ' \ \epsilon_{[\mathbf{t}]} \ ' \ \mathbf{t}_k \ ' \ B \ ' \ \zeta_{[\mathbf{t}]}$$

where $\mathbf{s}_k = \mathbf{t}_k$ *and:*

1. $\gamma_{[\mathbf{t}]} = \mathbf{t}_1, \ldots, \mathbf{t}_{j-1}$;
2. $\epsilon_{[\mathbf{t}]} = \mathbf{t}_{j+1}, \ldots, \mathbf{t}_{k-1}$;
3. $\zeta_{[\mathbf{t}]} = \mathbf{t}_{k+2}, \ldots, \mathbf{t}_n$;
4. *j is the greatest index lower than k such that:*
   (a) *$\mathbf{t}_j$ is not an auxiliary question, and*
   (b) *$\mathbf{t}_{j+1}, \ldots, \mathbf{t}_k$ is a sequence of questions;*
5. *B is a direct answer to the query $\mathbf{t}_k$.*

The main intuition behind the above corollary is that for each query $\mathbf{s}_k$, we can determine $\mathbf{t}_j$ $(j < k)$, such that $\epsilon_{[\mathbf{t}]}$ is a sequence of questions which precedes the answer B to a query $\mathbf{s}_k$. It is possible that $\epsilon_{[\mathbf{t}]}$ only consists of the query $\mathbf{s}_k$. If $\epsilon_{[\mathbf{t}]}$ consists of $n > 1$ elements, then it has to contain $n - 1$ auxiliary questions and exactly one query. If we consider the e-scenario from Fig. 1 and its first query $?\{p, \neg p\}$, then $\mathbf{t}_j$ is the initial question $(?\{(p \wedge q), \neg(p \wedge q)\})$ and $\epsilon_{[\mathbf{t}]}$ consists of $?\{p \wedge q, \neg p \wedge q, p \wedge \neg q, \neg p \wedge \neg q\}$ and $?\{p, \neg p\}$.

Fact 3.2. *Let Δ be a complete e-scenario for a question $\mathbf{s}_k$ relative to a set of d-wffs Y. Each path $\mathbf{g}$ of Δ can be represented as follows:*

$$\mathbf{s}_k \,'\, \mathsf{ids}_\Delta \,'\, \mathsf{faq}_\Delta \,'\, \delta_{[\mathbf{g}]} \,'\, C$$

where:

1. *$\mathbf{s}_k$ is the initial question;*
2. *ids_Δ is the initial declarative segment of the path;*
3. *faq_Δ is the first auxiliary question of the path;*
4. *$\delta_{[\mathbf{g}]}$ is the segment between first auxiliary question and the endpoint of the path;*
5. *C is the direct answer to $\mathbf{s}_k$.*

Consider the complete e-scenario from Fig. 1. The initial declarative segment of this e-scenario is empty and question $?\{p \wedge q, \neg p \wedge q, p \wedge \neg q, \neg p \wedge \neg q\}$ is the first auxiliary question of each path.

By Δ_B we mean the set of all paths of e-scenario Δ, which ends with the answer B to the initial question i.e. $\Delta_B = \{\mathbf{g} \in \Delta \mid \mathbf{g} = \mathbf{s}_k \,'\, \mathsf{ids}_\Delta \,'\, \mathsf{faq}_\Delta \,'\, \delta_{[\mathbf{g}]} \,'\, B\}$

Definition 3.1. *Let $\mathbf{t} \in \Sigma_{[\mathbf{s}, \mathbf{s}_k, B]}$ and $\mathbf{g} \in \Delta_B$.*

$$\mathbf{t} \otimes \mathbf{g} = \gamma_{[\mathbf{t}]} \,'\, \mathbf{t}_j \,'\, \mathsf{ids}_\Delta \,'\, \epsilon_{[\mathbf{t}]} \,'\, \mathbf{t}_k \,'\, \mathsf{faq}_\Delta \,'\, \delta_{[\mathbf{g}]} \,'\, B \,'\, \zeta_{[\mathbf{t}]}$$

Thus $\otimes$ is an operation which takes two e-derivations $\mathbf{t}$, $\mathbf{g}$ and combines them in a specific way. The following example illustrates how this operation works. Let

$$\mathbf{t} = \langle \Theta, ?\{p \wedge q, \neg(p \wedge q)\}, \neg(p \wedge q), \Theta' \rangle$$

where Θ/Θ' is the initial/final segment of e-derivation $\mathbf{t}$, which plays no role here and

$$\mathbf{g} = \langle ?\{p \wedge q, \neg(p \wedge q)\}, ?\{p \wedge q, \neg p \wedge q, p \wedge \neg q, \neg p \wedge \neg q\}, ?\{p, \neg p\}, \neg p, \neg(p \wedge q) \rangle$$

Now we want to paste e-derivation $\mathbf{g}$ into e-derivation $\mathbf{t}$. The result is the following e-derivation $\mathbf{t} \otimes \mathbf{g}$:

$$\mathbf{t} \otimes \mathbf{g} = \langle \Theta, ?\{p \wedge q, \neg(p \wedge q)\}, ?\{p \wedge q, \neg p \wedge q, p \wedge \neg q, \neg p \wedge \neg q\}, ?\{p, \neg p\}, \neg p, \neg(p \wedge q), \Theta' \rangle$$

Definition 3.2. *Let $\mathbf{s}$ be an e-scenario for question Q, $\mathbf{s}_k$ be a query of $\mathbf{s}$, and Δ be a complete e-scenario for a question $\mathbf{s}_k$.*

1. $\Sigma_{[\mathbf{s},\mathbf{s}_k]}^{\Delta_B,\otimes} = \{\mathbf{t} \otimes \mathbf{g} \mid \mathbf{t} \in \Sigma_{[\mathbf{s},\mathbf{s}_k,B]}$ *and* $\mathbf{g} \in \Delta_B\}$

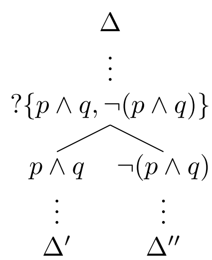
2. $\Sigma_{[\mathbf{s},\mathbf{s}_k]}^{\Delta,\otimes} = \bigcup_{B \in \mathsf{d}\mathbf{s}_k} \Sigma_{[\mathbf{s},\mathbf{s}_k]}^{\Delta_B,\otimes}$

Finally, the notion of *embedding* one scenario into another scenario can be introduced.

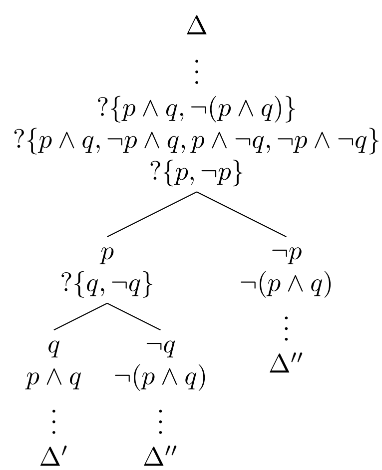
Definition 3.3 (Embedding)**.** *Let* $\mathbf{s}_k$ *be a query of a path* $\mathbf{s}$ *of an e-scenario* Σ*, and* Δ *be a complete e-scenario for question* $\mathbf{s}_k$*.*

$$\mathrm{EMB}(\Delta/\mathbf{s}, \mathbf{s}_k, \Sigma) = \widehat{\Sigma}_{[\mathbf{s},\mathbf{s}_k]} \cup \Sigma_{[\mathbf{s},\mathbf{s}_k]}^{\Delta,\otimes}$$

Let us look at the following example. We want to embed an e-scenario from Fig. 1 into the following e-scenario:



The result of embedding can be illustrated by the following tree:



The following theorem has been proven [40]:

Theorem 3.1 (Embedding Theorem)**.** *Let* Σ *be an e-scenario for question* Q *relative to a set of d-wffs* X*, and let* $\mathbf{s}_k$ *be a query of a path* $\mathbf{s}$ *of* Σ*. Let* Δ *be a complete e-scenario for question* $\mathbf{s}_k$ *relative to a set of d-wffs* Y*.* $\mathrm{EMB}(\Delta/\mathbf{s}, \mathbf{s}_k, \Sigma)$ *is an e-scenario for* Q *relative to* $X \cup Y$ *if the following conditions hold:*

1. $Y \cap \mathsf{d}Q = \emptyset$*, and*

2. *for each question* Q^* *of* Δ*:* $\mathsf{d}Q^* \neq \mathsf{d}Q$*.*

The operation of embedding produces an e-scenario from given e-scenarios Σ and Δ provided that the initial set of d-wffs of Δ does not contain an answer to the initial question of Σ. Moreover, the set of direct answers of each question of Δ can not be the same as the set of direct answers to the initial question of Σ.

3.2 Contraction

In what follows we will make an extensive use of the operation of contraction introduced in [40]. In particular, two of the operations proposed later, which eliminate redundancies in e-scenarios, are based on contraction.

Let Σ be an e-scenario, $\mathbf{s}$ be a path of Σ, and $\mathbf{s}_k$ be a query of $\mathbf{s}$. Note that an arbitrary element $\mathbf{t} = \mathbf{t}_1, \ldots, \mathbf{t}_u$ of $\Sigma_{[\mathbf{s},\mathbf{s}_{k+1}]}$ can be represented as follows (see Fact 3.1), where $\mathbf{t}_{k+1} = \mathbf{s}_{k+1}$:

$$\mathbf{t} = \gamma_{[\mathbf{t}]} \,'\, \mathbf{t}_j \,'\, \epsilon_{[\mathbf{t}]} \,'\, \mathbf{t}_k \,'\, \mathbf{t}_{k+1} \,'\, \zeta_{[\mathbf{t}]}$$

Note that if $\zeta_{[\mathbf{t}]}$ is not an empty string and some term $\zeta_{[\mathbf{t}]}$ is a question, then $\zeta_{[\mathbf{t}]} = \zeta^d_{[\mathbf{t}]} \,'\, \zeta^q_{[\mathbf{t}]}$, where $\zeta^d_{[\mathbf{t}]}$ is possibly an empty sequence of declarative formulas and $\zeta^q_{[\mathbf{t}]}$ begins with a question. We define an operation on $\mathbf{t}$:

Definition 3.4. *Let* $\mathbf{t} = \mathbf{t}_1, \ldots, \mathbf{t}_u \in \Sigma_{[\mathbf{s},\mathbf{s}_{k+1}]}$.

$$\ominus\mathbf{t} = \begin{cases} \gamma_{[\mathbf{t}]} \,'\, \mathbf{t}_j \,'\, \mathbf{t}_{k+1} & \text{if } u = k+1 \\ \gamma_{[\mathbf{t}]} \,'\, \mathbf{t}_j \,'\, \mathbf{t}_{k+1} \,'\, \zeta_{[\mathbf{t}]} & \text{if } u > k+1 \text{ and } \zeta_{[\mathbf{t}]} \text{ does not contain questions.} \\ \gamma_{[\mathbf{t}]} \,'\, \mathbf{t}_j \,'\, \mathbf{t}_{k+1} \,'\, \zeta^d_{[\mathbf{t}]} \,'\, \epsilon_{[\mathbf{t}]} \,'\, \zeta^q_{[\mathbf{t}]} & \text{if } u > k+1 \text{ and } \zeta_{[\mathbf{t}]} \text{ contains questions.} \end{cases}$$

In each of the cases above the query $\mathbf{t}_k$ is deleted. If the path $\mathbf{t}$ ends with the answer to $\mathbf{t}_k$ i.e. $\mathbf{t}_{k+1}$, we delete the segment $\epsilon_{[\mathbf{t}]}$. If the segment after $\mathbf{t}_{k+1}$, $\zeta_{[\mathbf{t}]}$ is non-empty and does not contain questions, we add it after $\mathbf{t}_{k+1}$ (as in the previous case the segment $\epsilon_{[\mathbf{t}]}$ is removed). If $\zeta_{[\mathbf{t}]}$ is non-empty and does contain questions, we add the segment $\zeta^d_{[\mathbf{t}]} \,'\, \epsilon_{[\mathbf{t}]} \,'\, \zeta^q_{[\mathbf{t}]}$ after $\mathbf{t}_{k+1}$ (the segment $\epsilon_{[\mathbf{t}]}$ is not deleted this time).

The following generalization is straightforward:

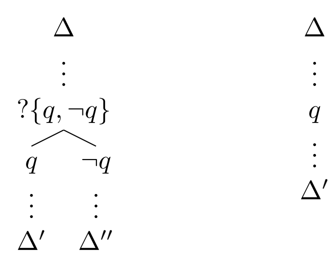
Definition 3.5. *Let* $\mathbf{s}$ *be a path of* Σ *and* $\mathbf{s}_k$ *be a query of* $\mathbf{s}$*:*

$$\Sigma^{\ominus}_{[\mathbf{s},\mathbf{s}_{k+1}]} = \{\ominus\mathbf{t} \mid \mathbf{t} \in \Sigma_{[\mathbf{s},\mathbf{s}_{k+1}]}\}$$

Definition 3.6 (Contraction)**.** *Let* $\mathbf{s}_k$ *be a query of a path* $\mathbf{s}$ *of an e-scenario* Σ*, and* $\mathbf{s}_{k+1}$ *be the direct answer to* $\mathbf{s}_k$ *occurring on* $\mathbf{s}$.

$$\mathrm{CTR}(\mathbf{s}_{k+1} \mid\mid \mathbf{s}, \mathbf{s}_k, \Sigma) = \widehat{\Sigma}_{[\mathbf{s},\mathbf{s}_k]} \cup \Sigma^{\ominus}_{[\mathbf{s},\mathbf{s}_{k+1}]}$$

Let us consider the following simplified example. Let Σ be an e-scenario and $\mathbf{s} = \langle s_1, \ldots, s_n\rangle$ be a path of Σ and let $s_i = ?\{q, \neg q\}$ (for some $1 < i < n$) be some query of $\mathbf{s}$. Before contraction, Σ can be represented as the tree on the left:



After contraction on q, an occurrence of a query $?\{q, \neg q\}$ is deleted, and an answer q is added at some specific place, according to the definition of the operation $\ominus$. The result of contraction may be represented as the tree on the right. Note that contraction may be applied with respect to $\neg q$ as well.

The central characteristic of contraction, which we will use later, is that the application of this operation to an e-scenario results (under certain conditions) in an e-scenario.

Theorem 3.2 (Contraction Theorem [40])**.** *Let* Σ *be an e-scenario for a question* Q *relative to a set of d-wffs* X*, and let* $\mathbf{s}_k$ *be a query of a path* $\mathbf{s}$ *of* Σ. $\mathrm{CTR}(\mathbf{s}_{k+1} \mid\mid \mathbf{s}, \mathbf{s}_k, \Sigma)$ *is an e-scenario for* Q *relative to* $X \cup \{\mathbf{s}_{k+1}\}$ *if*

1. $\mathbf{s}_{k+1} \notin \mathsf{d}Q$ *and*

2. $\widehat{\Sigma}_{[\mathbf{s},\mathbf{s}_k]} \neq \emptyset$ *or* $\Sigma_{[\mathbf{s},\mathbf{s}_k,,\mathbf{s}_{k+1}]}$ *involves at least two queries.*

The operation of contraction, when applied to an e-scenario, results in an e-scenario, when (according to Contraction Theorem) an answer to a query ($\mathbf{s}_{k+1}$) is not a direct answer to the initial question, and the complement of the set of all paths of Σ which has the formula $\mathbf{s}_k$ as the k-th term and the paths agree with path $\mathbf{s}$ as to previous items is not empty, or the set of all e-derivations which are identical to $\mathbf{s}$ up to the answer $\mathbf{s}_{k+1}$ to the query $\mathbf{s}_k$ contains at least two queries.

4 Redundancy elimination

We define a new set of functions such that each function transforms an e-scenario into another e-scenario. Some of these functions are based on contraction. Each function is thought of as eliminating a certain kind of redundancy, such as the repetition of queries on a certain path of an e-scenario. Some of these operations reduce the complexity of ESSs, and while this is not our primary goal, this feature may be beneficial in some applications. Let us start with a few auxiliary notions and the definition of a redundant path.

Definition 4.1 (e-equivalence)**.** *A question Q is* e-equivalent *to the question Q' (symbolically $Q =_e Q'$) iff*

1. $|\mathsf{d}Q| = |\mathsf{d}Q'|$

2. $\mathbf{Im}_{\mathcal{L}^?_{\mathsf{CPL}}}(Q, \emptyset, Q')$ *and*

3. $\mathbf{Im}_{\mathcal{L}^?_{\mathsf{CPL}}}(Q', \emptyset, Q)$*.*

It is said that two questions are equivalent iff they have the same number of direct answers and they e-imply each other. The notion of e-equivalence will be applied later only to simple yes–no questions so that the first condition of the definition will be satisfied trivially.

Fact 4.1. $=_e$ *is an equivalence relation between questions.*

By $Q^{=_e} = \{Q^* \mid Q =_e Q^*\}$ we mean the set of all questions e-equivalent to the question Q.

Definition 4.2 (d-equivalence)**.** *A d-wff A is* d-equivalent *to the formula A' (symbolically $A =_d A'$) iff*

1. $A \vdash_{\mathsf{CPL}} A'$ *and*

2. $A' \vdash_{\mathsf{CPL}} A$*.*

where $\vdash_{\mathsf{CPL}}$ is the syntactic consequence relation for CPL*.*

It is useful to have a general notion of equivalence which covers the notions of e-equivalence and d-equivalence.

Definition 4.3 (Equivalence)**.** *A formula B of $\mathcal{L}^?_{\mathsf{CPL}}$ is equivalent to the formula B' of $\mathcal{L}^?_{\mathsf{CPL}}$ (denoted by $B \mathrel{\widehat{=}} B'$) iff $B =_d B'$ or $B =_e B'$.*

The relation $\widehat{=}$ is an equivalence relation. Let us state the general definition of the concept of *redundant path*.

Definition 4.4 (Redundant path)**.** *Let $\mathbf{s} = \mathbf{s}_1, \ldots, \mathbf{s}_n$ be a path of an e-scenario Σ. A path $\mathbf{s}$ is said to be* redundant *with respect to $\mathbf{s}_i$ and $\mathbf{s}_k$ iff $\mathbf{s}$ can be represented as the following sequence:*

$$\gamma_{[\mathbf{s}]} \;'\; \mathbf{s}_i \;'\; \delta_{[\mathbf{s}]} \;'\; \mathbf{s}_k \;'\; \theta_{[\mathbf{s}]}$$

where:

Table 1: Types of repetitions of questions on a path of an e-scenario.

first occurrence	second occurrence	elimination yields an e-scenario?
query	*query*	yes (Sect. 4.1)
auxiliary question	*auxiliary question*	yes (Sect. 4.2)
query	*auxiliary question*	yes (Sect. 4.2)
auxiliary question	*query*	no

1. $\mathbf{s}_i \widehat{=} \mathbf{s}_k$
2. $\gamma_{[\mathbf{s}]} = \mathbf{s}_1, \ldots, \mathbf{s}_{i-1}$
3. $\delta_{[\mathbf{s}]} = \mathbf{s}_{i+1}, \ldots, \mathbf{s}_{k-1}$
4. $\theta_{[\mathbf{s}]} = \mathbf{s}_{k+1}, \ldots, \mathbf{s}_n$

If in the above definition $\mathbf{s}_i$ and $\mathbf{s}_k$ are questions, then the path $\mathbf{s} = \mathbf{s}_1, \ldots, \mathbf{s}_n$ is said to be *e-redundant* with respect to $\mathbf{s}_i$ and $\mathbf{s}_k$. If $\mathbf{s}_i$ and $\mathbf{s}_k$ are d-wffs, then the path $\mathbf{s} = \mathbf{s}_1, \ldots, \mathbf{s}_n$ is said to be *d-redundant* with respect to $\mathbf{s}_i$ and $\mathbf{s}_k$.

Note that $\mathbf{s}_i$ and $\mathbf{s}_k$ may not be the only terms which are equivalent in the above derivation $\mathbf{s}$. There may exist a term $\mathbf{s}^*$ in $\gamma_{[\mathbf{s}]}$, $\delta_{[\mathbf{s}]}$ or $\theta_{[\mathbf{s}]}$ which is equivalent to $\mathbf{s}_k$. For the sake of simplicity, in the reduction steps described later we will analyse only pairs of equivalent formulas in paths of scenarios. By recursively repeating these reduction steps we can effectively deal with an arbitrary number of equivalent formulas in a path.

When we claim that a question is repeated on some path of an e-scenario, what we mean is that this path contains two questions Q, Q' and $Q' \in Q^{=_e}$. Moreover, when we refer to the first occurrence of Q, we refer to some $Q^1 \in Q^{=_e}$, and when we refer to the second occurrence of Q, we refer to some $Q^2 \in Q^{=_e}$.

The types of repetitions of questions we consider here are presented in Table 1. The first column contains information about the type of the first occurrence of a repeated question Q on some path $\mathbf{s}$ of an e-scenario Σ (which may play the role of an auxiliary question or a query). The second column carries information about the type of the second occurrence of Q. Information about the effect of elimination of such repetitions as well as a reference to the appropriate section is contained in the third column.

For an arbitrary e-scenario Σ with a path containing two occurrences of a query, we define a procedure which transforms Σ into another e-scenario Σ', in which this particular repetition is eliminated. We define similar procedures for the pairs *auxiliary question–auxiliary question* and *query–auxiliary question*. Moreover, these procedures reduce the complexity of a given e-scenario, measured as the number of paths or the sum of lengths of all paths. However, the elimination of the type of repetition shown in the last row does not in general result in an e-scenario and when it does, the complexity may not be reduced. Therefore, this kind of elimination is not desired and we do not consider it in this work.

4.1 Co-occurrence of two queries

There are many kinds of redundant paths. Perhaps the most undesirable kind of redundancy is the redundancy with respect to a query of a path. For the sake of simplicity we consider only queries which have the form of a simple yes–no question[3], but it would be possible to extend the method to arbitrary queries. The effect of query repetition on a path $\mathbf{s}$ is that $\mathbf{s}$ contains a path with contradictory formulas.

Definition 4.5 (Query-redundant path). *A path* $\mathbf{s}$ *is q-redundant with respect to* $\mathbf{s}_i$ *and* $\mathbf{s}_k$ *iff* $\mathbf{s}$ *is an e-redundant path with respect to* $\mathbf{s}_i$ *and* $\mathbf{s}_k$*, and* $\mathbf{s}_i$, $\mathbf{s}_k$ *are queries.*

Definition 4.6 (Query-redundant e-scenario). *An e-scenario* Σ *is* q-redundant *if* Σ *contains a q-redundant path.*

[3] As proved in [40, ch. 12], each e-scenario can be transformed into an e-scenario which involves only yes–no questions as queries. Note that such a transformation may increase the complexity of the ESS.

It may happen that a set of d-wffs which occurs on a path $\mathbf{s}$ of an e-scenario Σ is inconsistent. If this is the case, we say that $\mathbf{s}$ is a *non-realisable path* of Σ. Inconsistency may be caused by the co-occurrence of two queries.

Definition 4.7 (Query-non-realisable path)**.** *A path* $\mathbf{s}$ *of* Σ *is said to be* query-non-realisable *with respect to* $\mathbf{s}_i$ *and* $\mathbf{s}_k$ *iff* $\mathbf{s}$ *can be represented as the following sequence:*

$$\gamma_{[\mathbf{s}]} \ ' \ \mathbf{s}_i \ ' \ \mathbf{s}_{i+1} \ ' \ \delta_{[\mathbf{s}]} \ ' \ \mathbf{s}_k \ ' \ \mathbf{s}_{k+1} \ ' \ \theta_{[\mathbf{s}]}$$

where $\mathbf{s}_i$ *and* $\mathbf{s}_k$ *are queries,* $\mathbf{s}_i =_e \mathbf{s}_k$ *and* $\mathbf{s}_{i+1} \neq_d \mathbf{s}_{k+1}$*.*

Definition 4.8. *An e-scenario* Σ *is (*query*)* non-realisable *(*nr*) iff* Σ *contains a (*query*) non-realisable path.*

Naturally, if a path $\mathbf{s}$ is query-non-realisable, then $\mathbf{s}$ is e-redundant. Let us state some useful facts.

Fact 4.2. *If a path* $\mathbf{s}$ *of* Σ *is non-realisable then it can be represented as*

$$\gamma_{[\mathbf{s}]} \ ' \ A \ ' \ \delta_{[\mathbf{s}]} \ ' \ B \ ' \ \theta_{[\mathbf{s}]}$$

where $A =_d \neg B$*.*

Fact 4.3. *There is no valuation which satisfies each d-wff formula on a non-realisable path, i.e. the set of formulas from a non-realisable path is 'inconsistent'.*

Let us now state a lemma that is crucial in the procedure of removing co-occurring queries. The main idea is the following: if some path $\mathbf{s}$ of an e-scenario Σ contains repeated queries $\mathbf{s}_i$, $\mathbf{s}_k$, and $\mathbf{s}$ is a query-non-realisable path, then we can always find a path which is *dual* to $\mathbf{s}$ (it has a different answer to the query $\mathbf{s}_k$) and it is not query-non-realisable with respect to $\mathbf{s}_i$ and $\mathbf{s}_k$.

Lemma 4.1. *If* Σ *is a non-realisable e-scenario and* $\mathbf{s}$ *is a query-non-realisable path of* Σ*:*

$$\mathbf{s} = \gamma_{[\mathbf{s}]} \ ' \ \mathbf{s}_i \ ' \ \mathbf{s}_{i+1} \ ' \ \delta_{[\mathbf{s}]} \ ' \ \mathbf{s}_k \ ' \ \mathbf{s}_{k+1} \ ' \ \theta_{[\mathbf{s}]}$$

(where $\mathbf{s}_i =_e \mathbf{s}_k$ *and* $\mathbf{s}_{i+1} \neq_d \mathbf{s}_{k+1}$*), then there exist a path of* Σ*, called* $\mathbf{s}^d$ *(a path dual to* $\mathbf{s}$ *with respect to* $\mathbf{s}_k$*) which can be represented as follows:*

$$\mathbf{s}^d = \gamma_{[\mathbf{s}]} \ ' \ \mathbf{s}^d_i \ ' \ \mathbf{s}^d_{i+1} \ ' \ \delta_{[\mathbf{s}]} \ ' \ \mathbf{s}^d_k \ ' \ \mathbf{s}^d_{k+1} \ ' \ \theta'_{[\mathbf{s}]}$$

where $\mathbf{s}_j = \mathbf{s}^d_j$ *for all* $j \leq k$ *and* $\mathbf{s}^d_{i+1} = \mathbf{s}^d_{k+1}$*.*

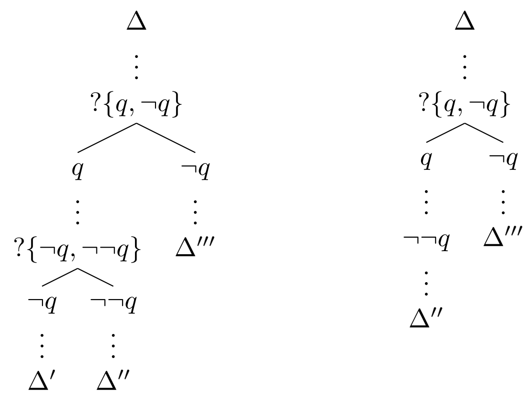
Note that if $\mathbf{s}$ is a query-non-realisable path, then $\mathbf{s}^d$ agrees with $\mathbf{s}$ up to the index k and then $\mathbf{s}$ has a formula $\mathbf{s}_{i+1}$ as an answer to the query $\mathbf{s}_k$. Now, using the lemma above, we are ready to define an operation which removes co-occurring queries.

Definition 4.9. *Let* Σ *be a non-realisable e-scenario,* $\mathbf{s}$ *be a path of* Σ*, which is query-non-realisable with respect to* $\mathbf{s}_i$ *and* $\mathbf{s}_k$*,* $\mathbf{s}^d$ *be the path dual to* $\mathbf{s}$ *with respect to* $\mathbf{s}_k$*. Then*

$$r^q(\mathbf{s}, \mathbf{s}_k, \Sigma) = \mathrm{CTR}(\mathbf{s}^d_{k+1} || \mathbf{s}^d, \mathbf{s}^d_k, \Sigma)$$

Let us look at the following example. On the left-hand side we have an e-scenario Σ with (at least two) repeated queries. We assume that the first query is answered by q. The result of applying operation

r^q is an e-scenario on the right-hand side. Query $?\{\neg q, \neg\neg q\}$ is deleted by performing contraction on the answer consistent with the previous one.



The use of the operation of contraction in Def. 4.9 assures that the complexity of Σ, measured as the number of paths, is reduced by the operation r^q and this operation has the following desired property:

Theorem 4.1. *If Σ is a non-realisable e-scenario, $\mathbf{s}$ is a path of Σ, which is query-non-realisable with respect to $\mathbf{s}_i$ and $\mathbf{s}_k$, and the conditions of* Contraction Theorem *hold, then $r^q(\mathbf{s}, \mathbf{s}_k, \Sigma)$ is an e-scenario.*

Proof. Follows from the Contraction Theorem 3.2 in Sect. 3.2. □

4.2 Co-occurrence of a query or an auxiliary question and an auxiliary question

Now we have to consider a situation when the second question is an auxiliary question and the first one is an arbitrary question.

Definition 4.10. *A path $\mathbf{s}$ of Σ is said to be* a-redundant *with respect to $\mathbf{s}_i$ and $\mathbf{s}_k$ $(i < k)$ iff $\mathbf{s}$ is an e-redundant path, $\mathbf{s}_i$ is an arbitrary question and $\mathbf{s}_k$ is an auxiliary question.*

Definition 4.11. *An e-scenario Σ is* a-redundant *iff Σ contains an a-redundant path.*

Now we define a simple function $\lambda_{\mathbf{s}}^{\mathbf{s}_k}$ which removes an element from a given path of an e-scenario.

Definition 4.12 (Shortening paths)**.** *Let $\mathbf{s} = \mathbf{s}_1, \mathbf{s}_2 \ldots, \mathbf{s}_n$ be a path of an e-scenario Σ and let $\mathbf{s}^*$ be a substring of $\mathbf{s}$, i.e. $\mathbf{s}^* = \mathbf{s}_2 \ldots, \mathbf{s}_n$.*

1. *$\lambda_{\mathbf{s}}^{\mathbf{s}_k} = \mathbf{s}$, if $\mathbf{s}$ is an empty sequence;*
2. *$\lambda_{\mathbf{s}}^{\mathbf{s}_k} = \mathbf{s}^*$ if $k = 1$ and $\mathbf{s}_1\ '\ \lambda_{\mathbf{s}^*}^{\mathbf{s}_{k-1}}$ otherwise.*

The notation $\lambda_{\mathbf{s}}^{\mathbf{s}_k}$ means 'the result of removing the k-th term from the path $\mathbf{s}$'. We need to generalise this operation to work on all paths which have $\mathbf{s}_k$ as the k-th term:

Definition 4.13 (Shortening e-scenarios)**.** *Let Σ be an e-scenario, $\mathbf{s}$ be an a-redundant path of Σ and $\Sigma_{[\mathbf{s},\mathbf{s}_k]}$ be the set of all paths that have $\mathbf{s}_k$ as the k-th term. Then*

$$\lambda_{\Sigma_{[\mathbf{s},\mathbf{s}_k]}}^{\mathbf{s}_k} = \{\lambda_{\mathbf{s}}^{\mathbf{s}_k} \mid \mathbf{s} \in \Sigma_{[\mathbf{s},\mathbf{s}_k]}\}$$

Intuitively, $\lambda_{\Sigma_{[\mathbf{s},\mathbf{s}_k]}}^{\mathbf{s}_k}$ denotes a set of paths such that the term $\mathbf{s}_k$ has been deleted from each of them. Using this operation we can define the operation which removes auxiliary questions.

Definition 4.14 (Removing auxiliary questions)**.** *Let Σ be an e-scenario, $\mathbf{s}$ be a path of Σ, which is a-redundant with respect to $\mathbf{s}_i$ and $\mathbf{s}_k$. Then*

$$r^{aux}(\mathbf{s}_k, \mathbf{s}, \Sigma) = \widehat{\Sigma}_{[\mathbf{s},\mathbf{s}_k]} \cup \lambda_{\Sigma_{[\mathbf{s},\mathbf{s}_k]}}^{\mathbf{s}_k}$$

If r^{aux} is applied in the appropriate context (specified in the theorem below), then we obtain an e-scenario as a result.

Theorem 4.2. *If Σ is an e-scenario, $\mathbf{s}$ is a path of Σ, which is a-redundant with respect to $\mathbf{s}_i$ and $\mathbf{s}_k$, then $r^{aux}(\mathbf{s}, \mathbf{s}_k, \Sigma)$ is an e-scenario.*

If a path $\mathbf{s}$ is a-redundant with respect to $\mathbf{s}_i$ and $\mathbf{s}_k$, then $\mathbf{s}_k$ is an auxiliary question and $\mathbf{s}_i$ is an arbitrary question (i.e. it is a query or an auxiliary question). In the proof of Theorem 4.2, we use the key observation that whenever $\mathbf{s}_k$ is used as an auxiliary question, the question $\mathbf{s}_i$ (which occurs earlier on a path) can be used instead of $\mathbf{s}_k$, and this does not change any dependencies (caused by the erotetic implication) which involved $\mathbf{s}_k$. As we do not use the operation of contraction to define r^{aux}, the number of paths of an e-scenario Σ does not decrease after the application of r^{aux}. Nevertheless, the complexity of Σ, measured as the sum of lengths of all of its paths, is reduced by the operation r^{aux}.

4.3 Co-occurrence of a query and an answer to this query

There is a possibility for an e-scenario to contain a path $\mathbf{s}$ such that somewhere on this path a declarative formula A occurs, and after this formula we have a query $?\{A, \neg A\}$. It is not reasonable to request information about A when this information is already present. This kind of redundancy is not mentioned in Table 1 due to the fact that this is not exactly a simple repetition. However, the effect of this kind of redundancy is a repetition of declarative formulas – a formula A and an answer to query $?\{A, \neg A\}$ which occurs after that query. We now define the operation which removes such cases of answered queries.

Definition 4.15. *Let $\mathbf{s} = \mathbf{s}_1, \ldots, \mathbf{s}_n$ be a path of an e-scenario Σ and let $\mathbf{s}_k$ be a query of $\mathbf{s}$. The query $\mathbf{s}_k$ is* non-informative on $\mathbf{s}$ *iff there exist $\mathbf{s}_i$, where $i < k$, such that $\mathbf{s}_i =_d \mathbf{t}$ and $\mathbf{t} \in \mathbf{ds}_k$.*

Intuitively, a query $\mathbf{s}_k$ is non-informative on a given path iff a formula, which is d-equivalent to some answer to that query, occurs on this path before $\mathbf{s}_k$. In this case, $\mathbf{s}_k$ is not needed.

Definition 4.16 (Superfluous path)**.** *A path $\mathbf{s}$ of Σ is said to be* superfluous *with respect to $\mathbf{s}_k$ iff $\mathbf{s}_k$ is an non-informative query on $\mathbf{s}$.*

Definition 4.17. *An e-scenario Σ is superfluous iff Σ contains a superfluous path.*

Once again we can use the operation of contraction in the following definition.

Definition 4.18 (Removing answered queries)**.** *Let Σ be an e-scenario, $\mathbf{s}$ be a superfluous path of Σ and $\mathbf{s}_k$ be a query of $\mathbf{s}$, which is non-informative and $\mathbf{s}_i =_d \mathbf{s}_{k+1}$ for $i < k$. Then*

$$r^{ans}(\mathbf{s}_k, \mathbf{s}, \Sigma) = \mathrm{CTR}(\mathbf{s}_{k+1} || \mathbf{s}, \mathbf{s}_k, \Sigma)$$

Thus the answered query $\mathbf{s}_k$ is removed by performing contraction on the direct answer to $\mathbf{s}_k$ which occurs earlier on a path (or a formula d-equivalent to this direct answer occurs earlier on a path). The following theorem follows from the Contraction Theorem 3.2 in Sect. 3.2).

Theorem 4.3. *If Σ is a superfluous e-scenario, $\mathbf{s}$ is a superfluous path of Σ, $\mathbf{s}_k$ is a query of $\mathbf{s}$, which is non-informative on $\mathbf{s}$ and the conditions of Contraction Theorem hold, then $r^{ans}(\mathbf{s}, \mathbf{s}_k, \Sigma)$ is an e-scenario.*

The set of functions introduced so far enables one to get rid of different types of redundancy which are sometimes hidden in generated e-scenarios. An e-scenario which can no longer be processed by means of introduced functions may be called an *optimal e-scenario.*

5 Automated generation and analyses of ESSs

A pragmatic approach to the decomposition of questions is typical to Inferential Erotetic Logic and Erotetic Search Scenarios. This approach raises the possibility of the employment of ESSs, as well as the knowledge that can be obtained from automated classification and optimisation, in various fields of data and language processing. ESSs can be applied in the domain of cooperative question-answer systems to empower interfaces for databases and other information systems [30]. Another area of application is the modelling of dependencies of questions in natural language dialogues such as information seeking dialogues and tutorial dialogues [29]. The potential usefulness of ESSs in dialogue analysis concerns the modelling of natural language with the formal framework of IEL, but also the employment of computational tools, such as those presented later in this section, to gain more general domain knowledge about interesting types of dialogues. Such knowledge can further facilitate the exploration of language patterns on the pragmatic level.

In previous theoretical sections we introduced the operations of embedding and contraction that enabled automatic generation of large sets of erotetic search scenarios. The set that we have obtained has been further processed by means of redundancy elimination procedures introduced in Sect.4. We proposed several evaluation criteria derived from the formal description of erotetic search scenarios [40]. Some of them, like *completeness* and *non-realisability*, were introduced in Sects. 3 and 4. To perform a more thorough analysis of ESSs, additional criteria such as *overflow* and *purity* will be introduced in this section; all the criteria are briefly summarized in Table 2 below.

As it has been demonstrated in [20], quantitative data generated in computational logic can reveal an implicit structure of logical problems. This is because computational tools facilitate processing of great amounts of instances of (usually complicated) problems and analysing these problems from many perspectives within a relatively short time and with little effort. Therefore, the computational apparatus stemming from artificial intelligence and from the implementations of logical algorithms is a suitable tool for application, development, evaluation and optimisation of IEL.

The theoretical framework founded by Winiewski, together with extensions introduced in Sects. 4 and 5, enabled us to apply the computational methodology in the area of Inferential Erotetic Logic. This approach was first proposed and employed in the domain of abductive hypotheses evaluation in [20]. The repertoire of computational tools is extended in this work to allow for the discovery of new knowledge about erotetic search scenarios.

In order to generate ESSs and to perform experiments based on the presented approach, a dedicated software – The Electronic Library of Erotetic Search Scenarios (ELESS) – has been created. This application is still in development; generated ESSs and quantitative data used in this work are published in the "Resources/Data" section at `http://intquestpro.wordpress.com`. The application is a part of a larger project devoted to the development of an electronic library of e-scenarios. As ELESS is capable of embedding and contraction operations and will contain even more means for automatic generation of e-scenarios due to the development of semantic tools, this library can be used to prepare a large set of e-scenarios. Since the library is ready, quantitative analyses of e-scenarios can be performed. These analyses take into account different ways e-scenarios can be evaluated, both qualitatively and quantitatively. ELESS contains e-scenarios generated manually and automatically (with the help of CESS, a Console application for Erotetic Search Scenarios). Each ESS is evaluated according to predefined criteria presented in Table 2. ELESS is also capable of cardinal evaluations (like *maximal depth*, *number of nodes* and others), but analyses of these statistics are not in the scope of this article.

5.1 Automatic generation of ESSs

The automatic generation of ESSs is achieved by embedding, contraction and transformation of scenarios into canonical and concise forms. The tools for performing these actions were implemented in Java programming language and were included into ELESS as one of its basic functionalities. For the purpose of experiments presented here, a set of ESSs was generated by applying the following procedure, for which the pseudocode (the *generate* function) is given in Alg. 4:

1. First, an initial set of 15 ESSs was manually generated. Five of these scenarios were the standard scenarios for CPL connectives. The remaining ten ESSs consisted of arbitrarily selected scenarios, some of which are for questions about tautologies[4], some contain initial premises (so they assume

[4]The initial question for such scenarios simply asks whether it is or it is not the case that the tautology holds. This kind of ESSs provides a strategy to answer such initial questions. There is a proof method called Synthetic Tableau

some prior knowledge), and some are complete (each answer to the initial question labels at least one leaf). All of these ten scenarios are realisable (do not contain inconsistent paths), non-redundant (do not contain paths with repeating formulas) and non-overflown (do not contain more than one answer to the initial question).

The initial set of e-scenarios is therefore composed of two parts. The first part consists of standard scenarios which are based on all the schemas available for CPC [40] and this part may be seen as basic building blocks (or even building schemas) of ESSs, as they concern simple yes–no questions about basic formulas with CPC connectives. The second part is more diversified and should allow for a verification of whether our procedures will be able to refine the initial set that contains such scenarios, to generate more efficient scenarios, or to gain new knowledge about them by using knowledge extraction procedures. This particular selection of the set of initial e-scenarios was made in order to test the refining abilities of the generation procedure. Standard scenarios and other simple e-scenarios were mixed with bigger, possibly redundant ones in order to increase the reliability of the obtained knowledge.

2. After that, new ESSs were generated by embedding each of the scenarios from the input set into each other with respect to every query of the main (the scenario in which another scenario was embedded) scenarios. The pseudocode for this part (the *embedContract* function) is given in Alg. 1. The embedding was performed only when it was permitted according to the embedding theorem (function $isEmbeddingAllowed(embedded\_scenario, main\_scenario, query\_of\_main\_scenario)$). Only one embedding operation at a time was allowed for each pair of the scenarios used, but it was possible to perform the procedure with respect to different queries of the main scenario in parallel. Before the embedding operation, the embedded scenario might have gone through renaming of the variables (function $rename(embedded\_scenario, query\_of\_main\_scenario)$), so that the principal question of that scenario and the embedding-candidate query would have been identical. The renaming procedure, however, was admissible only when it did not change the structural dependencies between formulas in the main scenario. After the embedding, if some of the formulas preceding the embedding point were identical with answers to the queries appearing beneath the embedding point, contractions were performed with respect to the answers that were already given before the queries and the embedding point, if permitted according to the contraction theorem (function $contract(scenario)$).[5] Finally, after all these procedures, all the repeating occurrences of the formulas already introduced earlier were removed from each path (function $delRedundant(scenario)$) – unless the latter occurrence was not a query or a leaf of the resulting scenario.

3. From all the scenarios generated in the previous step, new scenarios were generated by transforming considered ESSs into a concise form (function *makeConciseSet* in Alg. 2). After that, the input set was incrementally extended by the newly generated scenarios. The scenarios structurally equivalent to those already included in the input set were sieved out.

4. The previous step was repeated but with the transformation of the scenarios into a canonical form (function *makeCanonicalSet* in Alg. 3), so that all the initial premises in all the paths of an ESS appear before the first query.

5. Steps (2), (3) and (4) were repeated one more time starting with the set of scenarios generated up to this moment.

This procedure resulted in the generation of a set of 5671 scenarios.

5.2 Evaluation criteria for ESSs

The generated scenarios can be evaluated using numerical and nominal criteria. There are several criteria introduced in ELESS, but in this paper we investigate ordinal criteria because we focus on meta-logical

Method [33, 21, 20], derived from the ESSs concept and devoted to answer this kind of questions.

[5] The answers to a query are processed as a list. The first element from the list that is found before the query becomes a contraction point.

```
Input: set of e-scenarios S
Output: set of e-scenarios
R ← S
foreach s ∈ S do
    Q ← the set of queries of s
    foreach s' ∈ S \ {s} do
        foreach q ∈ Q do
            q' ← rename(s', q)
            if isEmbeddingAllowed(q', s, q) then
                s* ← embed(q', s, q)
                s* ← contract(s*)
                s* ← delRedundant(s*)
                R ← R ∪ {s*}
            end
        end
    end
end
return R
```

Algorithm 1: Function *embedContract*.

```
Input: set of e-scenarios S
Output: set of e-scenarios
R ← ∅
foreach s ∈ S do
    s* ← makeConcise(s)
    R ← R ∪ {s*}
end
return R
```

Algorithm 2: Function *makeConciseSet*.

```
Input: set of e-scenarios S
Output: set of e-scenarios
R ← ∅
foreach s ∈ S do
    s* ← makeCanonical(s)
    R ← R ∪ {s*}
end
return R
```

Algorithm 3: Function *makeCanonicalSet*.

```
Input: set of e-scenarios S
Output: set of e-scenarios
ITERATIONS ← 2
i ← 0
while i < ITERATIONS do                                  // step (5) in Sect. 5.1
    S ← S ∪ embedContract(S)            // step (2) in Sect. 5.1
    S ← S ∪ makeConciseSet(S)           // step (3) in Sect. 5.1
    S ← S ∪ makeCanonicalSet(S)         // step (4) in Sect. 5.1
    i ← i + 1
end
return S
```

Algorithm 4: Function *generate*.

Table 2: Criteria used to evaluate scenarios, the meaning of criteria values, and the number of scenarios characterized by each criterion value. Arrows near criteria names indicate the preference direction (↑ means maximisation, ↓ means minimisation).

Criterion	Values	Count
Completeness ↑	1 (*yes*) if leaves of an ESS contain all the answers to the principal question	1363
	0 (*no*) otherwise	4308
Conciseness ↑	2 (*yes*) if no path of an ESS contains a d-wff that is entailed by previous d-wffs	1459
	1 (*partial*) if there is at least one path containing such d-wffs, but not all paths have this property	1222
	0 (*no*) if all the paths contain d-wffs that can be deduced from previous d-wffs	2990
Canonicity ↑	2 (*yes*) if all the initial premises in all the paths of an ESS appear before the first query	2967
	1 (*partial*) if some of the paths have this property and some do not	2534
	0 (*no*) if every path contains the initial premises below the first query	170
Purity ↑	2 (*yes*) if each path of an ESS does not contain an initial premise	9
	1 (*partial*) if some paths do and some do not contain the initial premises	13
	0 (*no*) if all paths contain the initial premises	5649
Redundancy ↓	2 (*yes*) if all the paths contain more than one occurrence of the same formula	480
	1 (*partial*) if some but not all the paths contain repeating formulas	3270
	0 (*no*) if no path contains repeating formulas	1921
Realisability ↑	2 (*yes*) if each set of d-wffs obtained from every path is consistent	760
	1 (*partial*) if some (but not all) of the paths are such that the set of d-wffs obtained from each of these paths is inconsistent	4911
	0 (*no*) if each set of d-wffs obtained from every path is inconsistent	0
Overflow ↓	2 (*yes*) if all the paths contain more than one answer to the principal question	0
	1 (*partial*) if some (but not all) of the paths contain more than one answer to the principal question	4164
	0 (*no*) if no path contains more than one answer to the principal question	1507

high-level properties of ESSs and not on structural traits. Such criteria will be used to find satisfying (or optimal) scenarios. Seven ordinal criteria that have been used in this work are described in Table 2. Numbers accompanied by italicized names determine ranks of individual values.

The definitions of different types of redundancy can be found in Sect. 4. Realisability is defined in Sect. 4.1. The notion of a complete e-scenario is introduced in Def. 2.4.

5.3 Multi-criteria evaluation

Each of the generated scenarios has been evaluated using above-mentioned criteria. This enables finding ESSs desired for practical purposes, and allows further analyses of relationships between different kinds of scenarios. Finding the most interesting or efficient ESSs is interesting in itself and straightforward to achieve by either aggregating evaluation criteria or applying multi-criteria analyses. The latter approach based on the multi-criteria dominance relation [10, 9] is able to provide unbiased insight into the structure of the set of scenarios. Similarly as in [20], the evaluation of scenarios can be performed by imposing a preference order on values of each criterion. With such a preference order (determined by ranks described in the previous section), it is possible to select the most interesting scenarios by calculating which are the non-dominated ones.

Once the evaluation process was complete, data concerning scenarios were collected and represented in the form of a database table, where each row represents a particular scenario, each column represents a criterion (an "attribute" in data mining terminology), and cells contain individual values. Such data enables quantitative analyses of the ESSs and the application of the knowledge exploration algorithms.

The right column in Table 2 summarizes quantitative information about the generated ESSs. As it can be seen, some of the values are strongly under-represented. This is due to the applied generation procedure, primarily focused on obtaining non-redundant and realisable ESSs. What is more, the scenarios are generated by embedding and contraction performed on the set of initial scenarios, which are not representative for all the criteria values. It would be quite difficult to obtain a representative sample of all possible e-scenarios as there is practically no research reported yet on the representativeness of these constructs; this paper may be considered a starting point to conduct such research. However, as the initial set of ESSs contains simple forms of all standard scenarios (with the simplest conjunctions, alternatives, etc.), the obtained sample may be seen as intuitively basic, because standard scenarios that were used are basic themselves. These observations will have an impact on further interpretation of the results of the knowledge exploration process. It is also visible that the applied procedures did not generate scenarios that are fully not realisable and overflown, which is a desirable property. However, despite the efforts, some of the scenarios are partially not realisable, partially overflown or fully redundant. This observation indicates that there is still a room for improvements in the procedure. On the other hand, the not- and sub-optimal scenarios will still have some use at the knowledge exploration stage presented later.

The analysis of etymology of generated ESSs (middle part in Fig. 2) revealed that the initial scenarios $i1$–$i5$, $i11$, $i12$, $i15$ are rarely used in the production of new generations of ESSs. The first five of those are standard scenarios. Due to the fact that the rest of ESSs do not contain questions other than yes–no questions about simple facts, standard scenarios cannot be embedded anywhere which limits their "reproductive" capability. The next two scenarios decompose yes–no questions about tautologies and share the same fate as the standard scenarios. The last one is basically a non-concise version of scenario $i13$, and because it is processed later (due to the alphabetical order of processing), many of the e-scenarios descending from it would be the same as descendants of $i13$, so they have been sieved out. The most commonly used scenario is $i14$, which is decomposing a simple yes–no question. However, oppositely to ESSs $i7$, $i8$ and $i10$, which also concern simple yes–no questions, $i14$ makes use of variables rarely used in the other initial scenarios. This lowers the risk of breaching structural dependencies of $i14$ because of the variable renaming during embedding and contraction operations (see Sect. 5.1) and makes this scenario more versatile. The high applicability of $i14$ is further confirmed by its large contribution in the non-dominated set when the *purity* criterion is dropped, which is discussed in the next section.

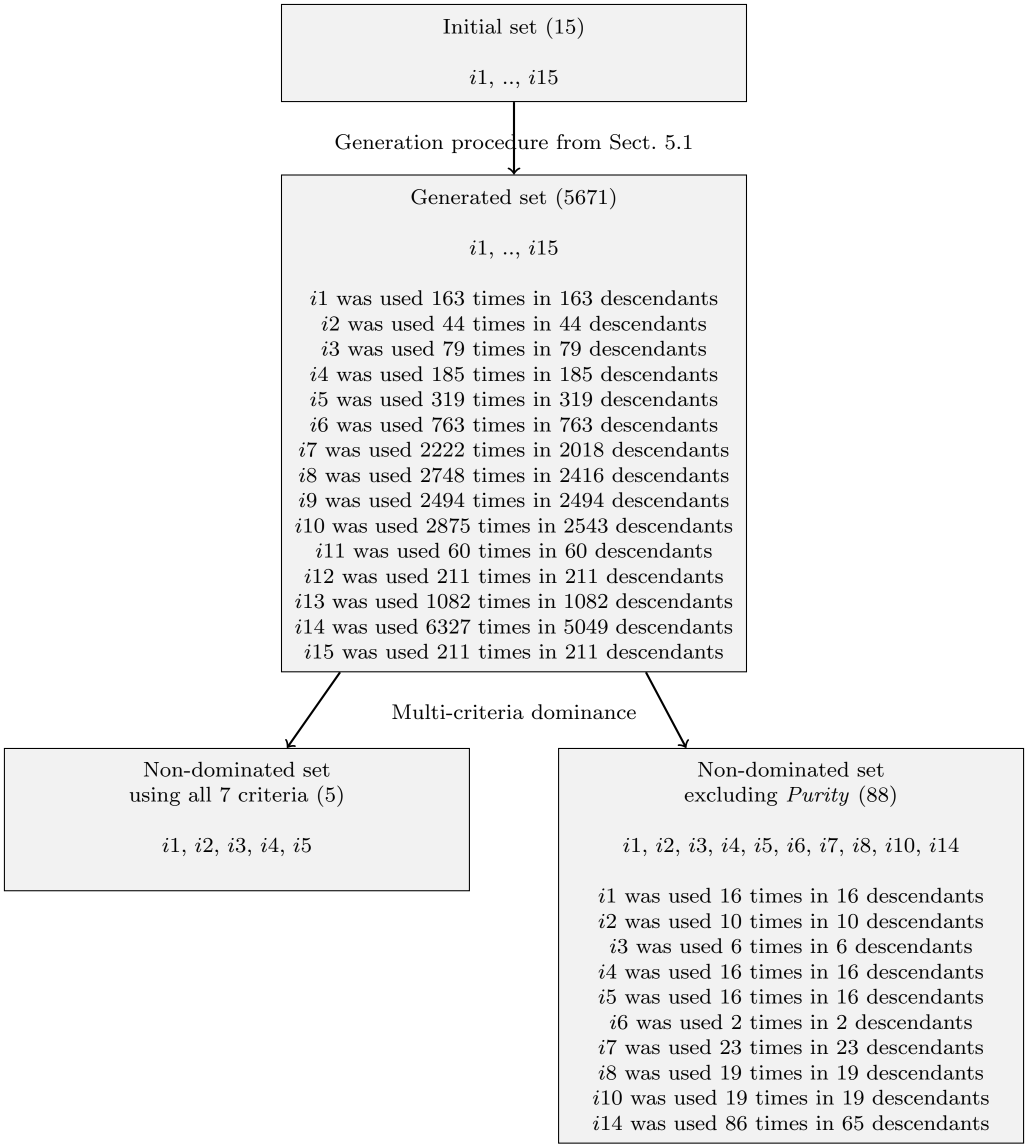


Figure 2: Summarised results of generation and optimisation of erotetic search scenarios.

5.4 Multi-criteria dominance relation

In this analysis we avoid aggregation of the criteria that evaluate e-scenarios, so there are no trade-offs introduced between the criteria. Multi-criteria analysis allows to identify scenarios that are better than other scenarios in any aspect. Relations that order values of individual criteria according to their preference are used to construct the multi-criteria dominance relation on scenarios in the following formal way: scenario Σ_1 is better than Σ_2 (i.e., Σ_1 dominates Σ_2), when Σ_1 is not worse (which means it is better or equally good) than Σ_2 on all criteria, and Σ_1 is strictly better than Σ_2 on at least one of these criteria. A scenario Σ that is not dominated by any other scenario in the set of all considered scenarios is called a non-dominated scenario, an efficient scenario, or a Pareto optimal scenario [10, 9].

The initial analysis employing the dominance relation defined above revealed that the set of non-dominated ESSs consists solely of all five standard scenarios for connectives of CPL, which are contained in the initial set of scenarios ($i1$–$i5$ in Fig. 2). This is not surprising as this class of scenarios is optimal (it maximises the values of all criteria). Moreover, the generation procedure from Sect. 5.1 does not contain

Table 3: Discordance between evaluation criteria imposed on the erotetic search scenarios generated according to the described procedure.

	canonicity	completeness	conciseness	overflow	purity	realisability	redundancy
canonicity	0.00	0.07	0.20	0.08	0.00	0.04	0.15
completeness	0.07	0.00	0.13	0.02	0.00	0.02	0.06
conciseness	0.20	0.13	0.00	0.12	0.00	0.07	0.16
overflow	0.08	0.02	0.12	0.00	0.00	0.00	0.04
purity	0.00	0.00	0.00	0.00	0.00	0.00	0.00
realisability	0.04	0.02	0.07	0.00	0.00	0.00	0.01
redundancy	0.15	0.06	0.16	0.04	0.00	0.01	0.00

any mechanism for removal of initial premises, hence the lack of balance is especially visible in the purity criterion.

It is not entirely clear if the inclusion of initial premises is a good or a bad trait of ESSs. On the one hand, the prior knowledge which is represented by the declaratives may be difficult or expensive to acquire, hence it may be desirable to minimise the number of initial premises. On the other hand, the most common practical problems likely depend on some initial knowledge, so the pure scenarios seem uninteresting – especially the standard ones which in a sense just explicate the logical knowledge about connectives.

There are a few interesting modifications of the way purity evaluation works. One is to turn the purity criterion into a numerical one by simply counting the number of declaratives, so that purity would be able to observe more subtle dependencies. Another way would be to introduce more interesting pure scenarios into the initial set. It would also be worthwhile to develop a mechanism that would intelligently remove the initial premises during the generation of new scenarios. A combination of these ideas could also be possible, and this will be our further work.

To gain more knowledge about the set of scenarios without the influence of purity we dropped this criterion, which resulted in an increase in the number of efficient ESSs up to 88.[6] Obviously, all the ESSs that were previously non-dominated retain their status as they are globally optimal, however new e-scenarios become non-dominated as well.

While the set of non-dominated scenarios when purity is ignored includes even more initial scenarios ($i6$, $i7$, $i8$, $i10$ and $i14$ in Fig. 2), the majority of the non-dominated set consists of scenarios generated at any stage of the procedure described in Sect. 5.1. Interestingly, all of the non-dominated generated scenarios are strict descendants of the non-dominated initial scenarios as it can be seen in the bottom-right panel in Fig. 2. This indicates that while the generation procedure is able to preserve the optimal criteria values of e-scenarios, it is not able to generate efficient scenarios from non-efficient ones, so the procedure can be further developed.

A few initial scenarios ($i9$, $i11$, $i12$, $i13$, $i15$) that are not present in the non-dominated set are not *complete*. The lack of completeness is especially unavoidable for $i11$ and $i12$, which concern initial yes–no questions about tautologies. Each path of those two ESSs ends with the same formula (the tautology). As the generation procedure cannot produce a complete scenario from incomplete scenarios (the procedure can only potentially break *completeness*), not surprisingly no descendant of incomplete scenarios entered the non-dominated set. On the other hand, in non-dominated e-scenarios all six criteria had optimal values, which is desirable and may also suggest a high agreement between criteria. The latter issue is examined in the next section.

5.5 Multi-criteria discordance analysis

The analysis of relationships between different criteria ranked according to the defined set of preferences is enabled by the discordance measure. The discordance measure is used to compare a pair of criteria, let us say C_1 and C_2, and it is calculated as a proportion of the number of pairs of different scenarios, for which one scenario is strictly better than the other on C_1 and strictly worse on C_2, to the number of all pairs scenarios. This measure gives an insight in how strongly criteria disagree in the given set of problems, and helps understand what is the relationship between different criteria when analysed globally by investigating all criteria simultaneously. The result presented in Table 3 suggests that the overall discordance between the criteria is relatively low: the maximal discordance is close to 20%. The reason for this may be that the procedure of ESSs generation was goal-directed, which ruled out many possible ESSs and made some of the criteria values under-represented. It may also mean that many of the criteria are interconnected and the selected order of preference emphasizes these relations, so that most criteria agree when evaluating e-scenarios generated by the procedure described in Sect. 5.1.

Before further analyses will be provided, note that the concept of the optimal scenario may differ depending on the actual context of the application of the ESSs, and therefore, on the order of preference of criteria values. Modifying the order of preferences will likely change discordances between the criteria.

The inspection of the results reveals that *purity* is the least conflicting criterion, its discordance with any of the other criteria not even approaching 1%. The reason for this is more than 99% of the generated EESs being *not* pure (this group is extremely over-represented) as seen in Table 2. Because of that, there is almost no room for disagreement with this criterion, as almost all of the e-scenarios share the same value of *purity*.

Realisability appears to be the second least conflicting criterion, despite the relatively low number of not redundant e-scenarios – less than 30% of all. The low discordance here is desirable and strengthens the view that if we want ESSs to be fine-grained, we should produce them realisable.

Redundancy, on the other hand, conflicts relatively often with the other criteria, especially with *canonicity* and *conciseness*. This demonstrates that there are still opportunities for further enhancements in the generation procedure. Redundancy may not invalidate a path of an e-scenario (like non-realisability does), but it may increase the cost of solving the problem. This increase may be caused by difficulty of acquiring information about the truth value of selected statements in the scenario or by the fact that the process needed to obtain such information uses much resources (e.g. time or computational power).

The most conflicting pair of criteria is *conciseness* and *canonicity* which is not surprising, as *conciseness* mildly disagrees with all the other criteria excluding *purity*. This observation indicates that in the future development of the ESSs generation procedure, more attention should be paid to the procedure of generation of the canonical, yet concise scenarios.

Interestingly, *overflow* moderately conflicts with *completeness*. A more thorough inspection revealed that all the conflicts were caused by partially overflown, complete ESSs and incomplete, non-overflown ones. This shows that the presented procedure can still be balanced to obtain more informative e-scenarios, where crucial pieces of information needed to solve the initial question are more efficiently distributed among paths of ESSs.

Yet another interesting observation concerns the disagreement between *conciseness* and *purity*, despite it being really low. While it seems intuitive that not every concise ESS has to be pure – a bit surprisingly, there were also scenarios that were pure but not concise. These were the scenarios that contained questions about tautologies, where it was possible to arrive at some formula without asking about it and without providing any prior knowledge, to arrive at the answer to the principal question.

In this research we have used all available scenarios and all the presented criteria in order to perform computations and to holistically show relationships between the criteria. However, in a practical real-life problem, a pre-selection stage can be added to initially sieve out unacceptable scenarios – for example, the partially not realisable ESSs are probably not useful for solving criminal or diagnostic problems.

The inspection of the criteria discordance matrix reveals some facts about the assessed scenarios and

[6] This behaviour is not inherent to the multi-criteria dominance relation. In general, the set of non-dominated ESSs does not have to change neither monotonically, nor anti-monotonically with respect to the number of criteria taken into consideration.

about the quality of the procedure designed to obtain them. This also encourages for further analyses of the relationships between different evaluations of the ESSs.

5.6 Mining relationships between criteria

In order to gain even more insight into the inter-criteria relationships, a knowledge exploration procedure has been employed. The analysis has been performed using the Apriori algorithm [1], which is a rule-mining tool used in many fields of data mining such as market basket analyses or searches for interdependencies in data. The output of this algorithm can be seen as a set of conditional sentences containing conjunctions of criteria values (i.e., scenario evaluations) in the antecedent and the consequent. There are two important parameters usually used to assess the quality of rules: *support* and *confidence*. In the presented case, *support* measures how many scenarios out of all the scenarios in the database are encompassed by a given rule, and *confidence* is a percentage value that indicates how often a rule is applied correctly. If a rule is $A \Rightarrow B$, then the support of this rule tells how often A and B occur together in the same instances of the database, and the confidence value tells how often A and B occur together out of all the cases where A occurred.

The most interesting rules are those which have a confidence value of 100%. This is because these rules are potential candidates for being facts interconnecting different classes of ESSs that describe meta-logical characteristics of ESSs. Obviously, as the collected data are not representative for all the possible ESSs, even the maximal value of confidence does not guarantee that such rules are provable. Therefore, at the present stage, these have to be revised by an expert (a logician), but in the near future we are going to employ an automatic prover that will analyse the relationships between scenarios. For the same reason, the support of rules is of little interest, because even rarely applied rules can actually describe meta-logical facts about classes which are not sufficiently represented in the data.

The Apriori algorithm [1] that was used here is implemented in Weka [13], an open-source data-mining software. The number of generated rules with a 100% confidence approached 5000, so here we only present a few selected ones. A high number of potentially interesting rules emphasizes the need for an automatic proving procedure.

The obtained rules indicate that knowledge gain in the process of data mining can be indeed interesting, however one has to be careful when analysing it, because not all of the rules are of general character.

Here we present three sample rules that reveal the characteristics of the generated ESSs. The number after the "criterion=value" expression is the number of ESSs having this criterion value. The number in parentheses is the confidence value of a rule.

- `purity=YES                9 ==> canonicity=YES          9  conf:(1.0)`
- `realisability=PARTLY 4911 ==> purity=NO             4911  conf:(1.0)`
- `completeness=FALSE   4308 ==> realisability=PARTLY 3939  conf:(0.91)`

The first rule is of general character and can be proved. All fully pure scenarios are also canonical (not only those present in the data). The reason for this is that fully pure scenarios do not contain any initial premises, which vacuously satisfies the condition for all the initial premises to be placed before the first query of the scenario in order for it to be in a canonical form.

The second rule that connects a partial lack of realisability with a total lack of purity – although maximally reliable on the basis of the collected data – is not a general proposition for Inferential Erotetic Logic. This example demonstrates that the selection of ESSs subjected to further analyses strongly influences the generality of a part of the extracted knowledge. This rule and the last one, even weaker in its generality (confidence < 1.0), do not deem the knowledge behind them useless. The lack of generality within a sensible range of confidence just limits the scope of applicability of those rules. If the set of scenarios generated by a certain procedure is the only thing that matters for practical reasons, than the knowledge built on this set should be sufficient as well.

On the other hand, consider the following rules:

- `redundancy=YES     480 ==> realisability=PARTLY  476  conf:(0.99)`

- `redundancy=PARTLY 3270 ==> realisability=PARTLY 3200  conf:(0.98)`

These two rules relate redundancy with respect to queries with non-realisability. Neither of these rules has the maximal confidence since there are many sources of redundancy in e-scenarios: declarative formulas, auxiliary questions and queries may be repeated as explained in Sect. 4. If declarative formulas are repeated, then an e-scenario may not contain a contradictory path. On the other hand, one can establish:

Proposition 5.1. *If an e-scenario Σ is q-redundant, then Σ is query-non-realisable.*

Proof. Assume Σ is q-redundant. Thus there exist $\mathbf{s}_i$ and $\mathbf{s}_j$, and $\mathbf{s}_i =_e \mathbf{s}_j$. Let $\mathsf{d}(\mathbf{s}_i) = \{A_1, A_2\}$ and $\mathsf{d}(\mathbf{s}_j) = \{B_1, B_2\}$. Assume also that $A_1 =_d B_1$ and $A_2 =_d B_2$ (from Def. 4.1). Therefore there exists a path $\mathbf{s}$ of Σ which contains A_1 and B_2. Since we consider only yes–no questions, $B_2 =_d \neg A_1$, $\mathbf{s}$ is a query-non-realisable path and Σ is a non-realisable e-scenario. Note that the assumption that $A_1 =_d B_2$ and $A_2 =_d B_1$ has the same consequence. □

This shows that adding more and more specific criteria values may be useful in discovering certain types of dependencies.

6 Conclusions

This article concerned automated generation and processing of *erotetic search scenarios* (ESSs). The formal account of ESSs and the operations that allow generation of new ESSs on the basis of already existing ones were presented. As these operations in their basic form do not guarantee that the resulting ESSs will be efficient, improvements and additional quality checks were proposed.

The ESSs generation procedure was implemented, and once a high number of potentially useful ESSs could be produced, several criteria were introduced to evaluate these scenarios. This allowed not only to identify efficient e-scenarios, but also to analyse discordances between evaluation criteria themselves. This analysis revealed that the selected criteria are relatively coherent, which is advantageous for further practical applications, as it is relatively easy to obtain an ESS which is acceptable according to several criteria at the same time.

Apart from the comparative analyses of the criteria, knowledge extraction was performed. This resulted in generation of a set of rules that describe dependencies between different criteria. The rules show either general dependencies provable with the help of a meta-logical apparatus, or strongly justified inter-dependencies that hold in the considered set of ESSs. Quantitative data acquired from these experiments provide general knowledge about e-scenarios. Such knowledge may be used for better understanding of e-scenarios, but also for development of heuristics for application of e-scenarios and for building new ESSs.

Alongside the results, this article presented a particular methodology (automatic generation of problems, discordance analyses, and knowledge extraction) that can be successfully applied to other logical structures, different from erotetic search scenarios (like it has already been done for abductive hypotheses [20]). However, even in the field of Inferential Erotetic Logic, further developments are in order – in particular, extending the base of initial ESSs, improving the procedure that generates scenarios (providing additional mechanisms to remove redundancy and inconsistencies) and adding new criteria. The last issue is related to automated generation of new knowledge concerning ESSs, which can be further exploited by designing an automatic prover for extracted rules.

While knowledge about ESSs is important from the meta-logical point of view, it can also be used to model human reasoning in dialogue [28]. Such models concern a specific situation or task, so their predictions could be compared with the real data (gained by employing experimental procedures or corpora analyses). Lastly, apart from the development of the ESSs library, future plans include introduction of the first-order language and calculus to ELESS. The final stage of the application work encompasses the use of electronic e-scenarios in independent applications such as databases managers [30] or in dedicated problem solvers.

References

[1] R. Agrawal and R. Srikant. 1994. Fast algorithms for mining association rules. In *Proceedings of 20th International Conference on Very Large Data Bases*, Vol. 1215. 487–499.

[2] C. Baskent. 2016. *Perspectives on Interrogative Models of Inquiry. Developments in Inquiry and Questions*. Logic, Argumentation & Reasoning, Vol. 8. Springer, Heidelberg/New York/Dordrecht/London.

[3] N. D. Belnap. 1969. Åqvists corrections-accumulating question-sequences. In *Philosophical logic*. Springer, 122–134.

[4] N. D. Belnap and Th. P. Steel. 1976. *The Logic of Questions and Answers*. Yale University Press, New Haven.

[5] Sz. Chlebowski and D. Leszczyńska-Jasion. 2015. Dual Erotetic Calculi and the Minimal LFI. *Studia Logica* 103, 6 (2015), 1245–1278.

[6] I. Ciardelli, J. Groenendijk, and F. Roelofsen. 2013. On the Semantics and Logic of Declaratives and Interrogatives. *Synthese* 192 (2013), 1689–1728.

[7] I. Ciardelli and F. Roelofsen. 2011. Inquisitive Logic. *Journal of Philosophical Logic* 40 (2011), 55–94.

[8] I. Ciardelli and F. Roelofsen. 2015. Inquisitive dynamic epistemic logic. *Synthese* 192, 6 (2015), 1643–1687.

[9] M. Doumpos and E. Grigoroudis. 2013. *Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications*. Wiley.

[10] M. Ehrgott. 2006. *Multicriteria Optimization*. Springer.

[11] J. Ginzburg. 1995. Resolving questions, I. *Linguistics and philosophy* 18, 5 (1995), 459–527.

[12] J. Ginzburg. 2011. Questions: logic and interactions. In *Handbook of logic and language*. Elsevier, 1133–1146.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.* 11, 1 (Nov. 2009), 10–18. `DOI:http://dx.doi.org/10.1145/1656274.1656278`

[14] Y. Hamami. 2014. The interrogative model of inquiry meets dynamic epistemic logics. *Synthese* 192, 6 (2014), 1609–1642.

[15] C. L. Hamblin. 1958. Questions. *The Australian Journal of Philosophy* 36 (1958), 159–168.

[16] David Harrah. 2002. The logic of questions. In *Handbook of philosophical logic*. Springer, 1–60.

[17] J. Hintikka. 1981. On the logic of an interrogative model of scientific inquiry. *Synthese* 47, 1 (1981), 69–83.

[18] J. Hintikka. 1999. *Inquiry as Inquiry: A Logic of Scientific Discovery*. Kluwer, Dordrecht/Boston/London.

[19] J. Hintikka. 2007. *Socratic Epistemology: Explorations of Knowledge-Seeking by Questioning*. Cambridge University Press, Cambridge.

[20] M. Komosinski, A. Kups, D. Leszczyńska-Jasion, and M. Urbański. 2014. Identifying efficient abductive hypotheses using multi-criteria dominance relation. *ACM Transactions on Computational Logic* 15, 4 (2014), 28:1–28:20. `DOI:http://dx.doi.org/10.1145/2629669`

[21] M. Komosinski, A. Kups, and M. Urbanski. 2012. Multi-criteria evaluation of abductive hypotheses: towards efficient optimization in proof theory. In *Proceedings of the 18th International Conference on Soft Computing*. Brno, Czech Republic, 320–325.

[22] T. Kubiński. 1980. *An outline of the logical theory of questions*. Akademie-Verlag, Berlin.

[23] T. A. F. Kuipers and A. Wiśniewski. 1994. An erotetic approach to explanation by specification. *Erkenntnis* 40, 3 (1994), 377–402.

[24] D. Leszczyńska. 2004. Socratic Proofs for some Normal Modal Propositional Logics. *Logique et Analyse* 47, 185–188 (2004), 259–285.

[25] D. Leszczyńska-Jasion. 2008. The Method of Socratic Proofs for Modal Propositional Logics: K5, S4.2, S4.3, S4M, S4F, S4R and G. *Studia Logica* 89, 3 (2008), 371–405.

[26] D. Leszczyńska-Jasion. 2013. *Erotetic search scenarios as families of sequences and erotetic search scenarios as trees: Two different, yet equal accounts*. Technical Report. Adam Mickiewicz University.

[27] D. Leszczyńska-Jasion, M. Urbański, and A. Wiśniewski. 2013. Socratic Trees. *Studia Logica* 101, 5 (2013), 959–986.

[28] P. Łupkowski. 2015. Question dependency in terms of compliance and erotetic implication. *Logic and Logical Philosophy* 24, 3 (2015), 357–376. `http://apcz.pl/czasopisma/index.php/LLP/article/view/LLP.2015.002`

[29] Paweł Łupkowski. 2016. Logic of Questions in the Wild. Inferential Erotetic Logic in Information Seeking Dialogue Modelling. (2016).

[30] P. Łupkowski and D. Leszczyńska-Jasion. 2014. Generating cooperative question-responses by means of erotetic search scenarios. *Logic and Logical Philosophy* 24, 1 (2014), 61–78. `http://apcz.pl/czasopisma/index.php/LLP/article/view/LLP.2014.017`

[31] M. Peliš and O. Majer. 2009. Logic of questions and public announcements. In *Logic, Language, and Computation*. Springer, 145–157.

[32] D. J. Shoesmith and T. J. Smiley. 1978. *Multiple-conclusion logic*. Cambridge University Press, Cambridge.

[33] Mariusz Urbański. 2001. Remarks on Synthetic Tableaux for Classical Propositional Calculus. *Bulletin of the Section of Logic* 30, 4 (2001), 194–204.

[34] M. Urbański and A. Wiśniewski. 2016. On Search for Law-Like Statements as Abductive Hypotheses by Socratic Transformations. In *Perspectives on Interrogative Models of Inquiry*. Springer, 111–127.

[35] J. van Benthem and S. Minica. 2012. Toward a Dynamic Logic of Questions. *Journal of Philosophical Logic* 41 (2012), 633669.

[36] A. Wiśniewski. 1995. *The Posing of Questions: Logical Foundations of Erotetic Inferences*. Kluwer, Dordrecht/Boston/London.

[37] A. Wiśniewski. 1999. Erotetic logic and explanation by abnormic hypotheses. *Synthese* 120, 3 (1999), 295–309.

[38] A. Wiśniewski. 2003. Erotetic Search Scenarios. *Synthese* 134, 3 (2003), 389–427.

[39] A. Wiśniewski. 2004. Socratic Proofs. *Journal of Philosophical Logic* 33 (2004), 299326.

[40] A. Wiśniewski. 2013. *Questions, Inferences and Scenarios*. College Publications, London.

[41] A. Wiśniewski and V. Shangin. 2006. Socratic Proofs for Quantifiers. *Journal of Philosophical Logic* 35, 2 (2006), 147–178.