

Identifying efficient abductive hypotheses using multi-criteria dominance relation

Maciej Komosinski
Adam Kups
Dorota Leszczyńska-Jasion
Mariusz Urbański

Abstract

In this article, results of the automation of an abductive procedure are reported. This work is a continuation of our earlier research [21], where a general scheme of the procedure has been proposed. Here, a more advanced system developed to generate and evaluate abductive hypotheses is introduced. Abductive hypotheses have been generated by the implementation of the Synthetic Tableau Method. Before the evaluation, the set of hypotheses has undergone several reduction phases. To assess usefulness of abductive hypotheses in the reduced set, several criteria have been employed. The evaluation of efficiency of the hypotheses has been provided by the multi-criteria dominance relation. To comprehend the abductive procedure and the evaluation process more extensively, analyses have been conducted on a number of artificially generated abductive problems.

1 Introduction

Abductive reasoning aims at making sense of puzzling phenomena [14]. It has been argued that abduction underlies all kinds of interpretative reasoning [24, 35], ranging from diagnostic reasoning and anomaly-driven scientific explanation-seeking to natural language understanding and empathy. Its general schema is this: from an observation A and the known rule *if H , then A* , infer H (cf. [32, 5.189]). However, this schema may be elaborated in detail in different ways, which lead to different models of abduction. Here we follow the algorithmic point of view, according to which an abductive hypothesis H (or an abducible, as we shall also call it) “is legitimately dischargeable to the extent to which it makes it possible to prove (or compute) from a database a formula not provable (or computable) from it as it is currently structured” [11, p. 88].

In such a setting, an abductive problem may be expressed in terms of (logical) entailment: given the set of formulas X (we shall call this set a *database*) and a formula A such that X does not entail A (an *abductive goal*, or simply a *goal*), find a formula H (an *abductive hypothesis* or an *abducible*), such that X and H together entail A . Thus abduction understood in algorithmic terms amounts to filling a (deductive, or computational) gap between X and A .

There are four ingredients of the algorithmic account of abduction: (i) a basic logic (which determines the language of specification of A , H and X), (ii) a proof method (which determines the exact mechanics of the procedure of generation of abducibles), (iii) a hypotheses generation mechanism (which determines the way the chosen proof method is applied in order to generate abducibles), and (iv) an implementation of criteria for comparative evaluation of different abducibles.

Various procedures for generation of abducibles have been proposed so far; some are designed for Classical Propositional Calculus (CPC for short) [1], others for more sophisticated propositional logics [29, 26] or for first-order logic [16, 25, 28]. Those procedures, which are defined in a strictly logical

The final version of this paper appeared in *ACM Transactions on Computational Logic* 15(4), pages 28:1–28:20, 2014. <http://dx.doi.org/10.1145/2629669>

setting, use, for example, the proof methods of analytic tableaux [1, 26, 25], of sequent calculi [26, 25], or the dynamic proof method of adaptive logics [29, 28]. In this paper we will consider an implementation of a procedure generating abductive hypotheses for CPC.¹ The choice of this basic logic is motivated by its simplicity. We have implemented an abductive procedure based on the Synthetic Tableaux Method, STM [36] (Sect. 2), using the scripting programming language (Sect. 3). There are no other accounts of the problem using STM. This work is a continuation of our earlier work [21], with a number of significant innovations introduced into the procedure. The differences between the previous and the current approaches are reported in Sect. 3.4.

From the algorithmic point of view, implementation of a procedure that generates abductive hypotheses is a relatively simple task. The challenging part is the computationally tractable evaluation of hypotheses, as the space of relevant formulas may grow exponentially with the increase of the cardinality of the database and the number of distinct atomic formulas occurring in it (cf. [4, 10]). We shall focus on the fourth ingredient: criteria for comparative evaluation of abductive hypotheses. The set of criteria against which abducibles are evaluated comprises usually some combinations of relevance, complexity and consistency (provided that the basic logic is consistency-sensitive). To these, we add additional criteria of significance and operational complexity. We shall consider dominance relations based on different subsets of the criteria just mentioned (Sect. 4).

2 Logical basis

2.1 The Synthetic Tableaux Method

STM is a model-seeking and a proof method. It was developed in detail in [36] as a decision procedure for Classical Propositional Calculus (CPC) and for some non-classical logics (see also [37] and [38]). The fundamental ideas underlying STM can be traced back to L. Kalmár's proof of the completeness of the CPC. Roughly speaking, a synthetic tableau for a formula B is defined as a family of interconnected derivations of B and/or $\neg B$ (the so-called *synthetic inferences of $B/\neg B$*) on the basis of suitably defined sets of certain basic constituents. The choice of the sets of constituents is logic-dependent; in the case of CPC, they are consistent sets of literals occurring in $Sub(B)$ or their negations.² The formulas occurring in a synthetic tableau for a formula B satisfy the subformula condition: they can only be elements of $Sub(B)$ or their negations. As a result, all the compound formulas occurring in a certain synthetic inference \mathbf{s} of a formula B are elements of $Sub(B)$ or their negations, derived on the basis of the literals of \mathbf{s} (which are also elements of $Sub(B)$ or their negations). The relevant derivability relation is defined by means of the following ten rules (of a purely synthesizing character):

$$\begin{array}{ccccc}
 \mathbf{r}_\rightarrow^1 \frac{\neg\alpha}{\alpha \rightarrow \beta} & \mathbf{r}_\rightarrow^2 \frac{\beta}{\alpha \rightarrow \beta} & \mathbf{r}_\rightarrow^3 \frac{\alpha, \neg\beta}{\neg(\alpha \rightarrow \beta)} & \mathbf{r}_\vee^1 \frac{\alpha}{\alpha \vee \beta} & \mathbf{r}_\vee^2 \frac{\beta}{\alpha \vee \beta} \\
 \mathbf{r}_\vee^3 \frac{\neg\alpha, \neg\beta}{\neg(\alpha \vee \beta)} & \mathbf{r}_\wedge^1 \frac{\neg\alpha}{\neg(\alpha \wedge \beta)} & \mathbf{r}_\wedge^2 \frac{\neg\beta}{\neg(\alpha \wedge \beta)} & \mathbf{r}_\wedge^3 \frac{\alpha, \beta}{\alpha \wedge \beta} & \mathbf{r}_\neg \frac{\alpha}{\neg\neg\alpha}
 \end{array}$$

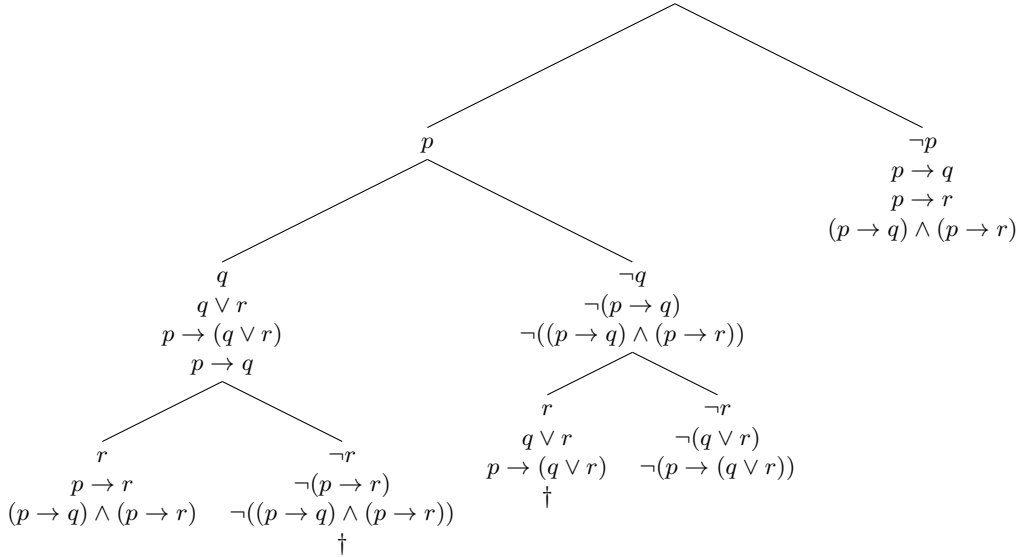
STM is a semantically motivated tableau method, yet it is based on a direct reasoning. E.g., in the case of CPC, each synthetic inference may be viewed as a syntactic codification of the calculation of the

¹We assume the usual notion of a CPC-formula, in a language with \neg (negation), \vee (disjunction), \wedge (conjunction) and \rightarrow (implication) as primitive connectives (equivalence \equiv may be defined as usual, in terms of implication and conjunction). We assume also the usual notion of a subformula of a given formula. We use $Sub(B)$ to represent the set of all the subformulas of a formula B and $Sub(X)$ to represent the union of sets of subformulas of all the elements of a set X of formulas. By a *literal* we mean a propositional variable or the negation of a propositional variable. A formula which is not a literal is called a *compound* formula. If for two literals l_1, l_2 one of them is of the form p_i and the other is of the form $\neg p_i$, then we say that l_1 and l_2 are *complementary*. If $X = \{A_1, \dots, A_n\}$ is a finite set of formulas, then by $\bigwedge X$ we mean the conjunction of its elements: $A_1 \wedge \dots \wedge A_n$. Similarly, by $\bigvee X$ we mean the disjunction: $A_1 \vee \dots \vee A_n$.

²Precise definitions of synthetic inferences and synthetic tableaux can be found in [36] and [39].

value of B under a valuation constrained by the values of literals present in this synthetic inference. Thus it also follows that the complexity of STM for CPC is bounded by the complexity of the truth-tables method.

For the purpose of modelling abductive reasoning, the following, more general notion of a synthetic tableau is introduced. A *synthetic tableau for a derivation of a formula B on the basis of a set of formulas Y* is defined as a family of interconnected derivations (again, *synthetic inferences of B on the basis of Y*) of elements of $Y \cup \{B\}$ or their negations, on the basis of consistent sets of literals occurring in $Sub(Y \cup \{B\})$ or their negations. A synthetic inference s of a formula B on the basis of the set $Y = \{D_1, \dots, D_n\}$ is called a *success* iff either there exists at least one $D_i \in Y$ such that $\neg D_i$ is an element of s , or B is an element of s . It is called a *failure* iff all of the following are elements of s : $D_1, \dots, D_n, \neg B$, that is, iff it contains every formula from the database and negation of the abductive goal. A set Y entails a formula B ($Y \models B$) iff there exists a tableau Ω for the derivation of B on the basis of Y such that each element of Ω is a success.³ All the details and metatheorems concerning STM can be found in [39]. Here we offer an example that shall give an idea of how the STM works. This is a synthetic tableau Ω_1 for the derivation of $B = (p \rightarrow q) \wedge (p \rightarrow r)$ on the basis of $Y = \{p \rightarrow (q \vee r)\}$:



Consider the leftmost synthetic inference (call it s) of the tableau Ω_1 . Literals p, q, r are introduced as basic constituents occurring in $Y \cup \{B\}$. The order in which they are introduced is arbitrary. (However, it may be of importance with respect to complexity of a tableau.) Each time a literal l_1 is introduced as the $i + 1$ st term of s a new synthetic inference is created, which is identical with s up to their i th terms (if there are any) and with a literal complementary to l_1 as its $i + 1$ st term. The compound formulas as introduced by applications of the following rules: $q \vee r$ results from q by \mathbf{r}_\vee^1 , $p \rightarrow (q \vee r)$ results from $q \vee r$ by \mathbf{r}_\rightarrow^2 , $p \rightarrow q$ results from q by \mathbf{r}_\rightarrow^2 , $p \rightarrow r$ results from r by \mathbf{r}_\rightarrow^2 , and $(p \rightarrow q) \wedge (p \rightarrow r)$ results from $(p \rightarrow q)$ and $(p \rightarrow r)$ by \mathbf{r}_\wedge^3 .

Tableau Ω_1 consists of five synthetic inferences; two of them, marked by \dagger , are failures (and their existence proves that the set $\{p \rightarrow (q \vee r)\}$ does not entail the formula $(p \rightarrow q) \wedge (p \rightarrow r)$). The remaining three are successes.

³A tableau Ω is a proof of a formula B (that is, of the fact that $\emptyset \models B$) iff each element of Ω is a synthetic inference of B . Intuitively it can be said that a formula B is proved iff all the possible attempts at synthesizing B or $\neg B$ on the basis of the consistent sets of their subformulas (with the sets of basic constituents of B or their negations interpreted as representing “initial conditions” or “basic assumptions”) lead to B .

In a given synthetic inference s , all the formulas except literals are introduced as derived on the basis of the preceding formulas. As a result, all compound formulas of s are derived on the basis of the set of literals occurring in s . It may happen, however, that not all literals in a given synthetic inference are needed in order to derive compound formulas occurring in this inference. Consider the following sequence:

$$s_1 = \neg p, p \rightarrow q, q, p \rightarrow r, \neg r, (p \rightarrow q) \wedge (p \rightarrow r)$$

This is a slightly modified rightmost synthetic inference of the tableau Ω_1 . Notice that q and $\neg r$ are in a sense superfluous in s_1 : all the compound formulas are derived on the basis of $\neg p$. In other words, only $\neg p$ is an *entangled* literal in s_1 – that is a literal which is relevant for s_1 being a success or a failure.⁴

2.2 Modelling abduction by STM

Given a set of premises (a database) X and a formula A such that X does not entail A (an abductive goal), the procedure of generating abducibles is defined as follows:

- (T1) A synthetic tableau Ω for a derivation of A on the basis of X is generated.
- (T2) Suppose that Ω has k failures.⁵ For each failure s_i of Ω , the entangled literals C_1^i, \dots, C_n^i of s_i are extracted.
- (T3) For each failure s_i , a formula $D_i = \neg(C_1^i \wedge \dots \wedge C_n^i)$ is built. Next, for the set $E = \{D_1, \dots, D_i, \dots, D_k\}$ a conjunction $\bigwedge E$ is created.
- (T4) The formula $\bigwedge E$ is transformed into the disjunctive normal form $F_1 \vee \dots \vee F_m$. Each disjunct F_j of E is an abducible for the formula A with respect to X .

Finally, for each non-empty subset S of $\{F_1, \dots, F_m\}$, the disjunction $\bigvee S$ is also an abductive hypothesis for A with respect to X . Observe that we could also consider any conjunction of the abducibles obtained so far as an abducible. However, we will not do this for the sake of minimality of the abductive hypotheses.

As mentioned above, implementation of a procedure that generates abductive hypotheses is a relatively simple task. As a result, the choice of proof method, however important, is not crucial. We decided to employ STM as we wanted to avoid some well-known problems of resolution techniques, albeit this comes at a computational price, of which we are aware.⁶

3 Generation and reductions of the set of abductive hypotheses

3.1 Implementation of the synthesizing algorithm

Details of the synthesizing algorithm were given in [21]; in this section, only a concise description of the implementation of the STM will be provided, along with the explanation of its application in solving abductive problems. The implementation was performed using the scripting language, FramScript, which is a part of the Framsticks platform [23]. The platform is especially suitable for designing, testing, and optimizing connectionist [18] and multi-agent [22] models; it has been previously used to optimize fuzzy logic controllers [13]. While other programming languages could be chosen at this stage,

⁴A precise definition of entanglement is given in [39].

⁵Let us add that since X does not entail A , Ω must contain – by soundness of STM – at least one failure, thus we know that $k \geq 1$.

⁶In general, STM fulfills criteria for a system amenable to computerized proof search [3, p. 18]: (a) proof search methods are efficient and do not require too many “arbitrary” choices, and (b) proof lengths are not excessively long. From a computational point of view, STM is at least as good as the truth-tables method, thus STM may compete with analytic methods at least in the case of certain well-known classes of cases (see [6]).

this particular environment has been used because of the available library of functions for processing logic formulas [20], support for distributed or connectionist logic models, and because of support for various optimization algorithms that will be useful in further experiments.

The implementation of the Synthetic Tableau Method builds a binary tree of logical formulas. Before execution of the synthesizing function, all the formulas of an abductive problem are decomposed into subformulas. From the set of subformulas the set of all variables is acquired. The list of variables is then passed to the function that builds the binary tree. Each binary split (including the initial branching) is generated by taking the variables from the passed list of variables. The tree is built in a symmetrical manner: each branching of the same depth splits into identical pairs of complementary literals. The order of inserted literals depends on the order of the variable list, which is in turn determined by the order of generated subformulas. The new formulas appear on a branch as a result of applying every possible rule to the formulas already present in the branch, however, only the formulas from the initially generated set of subformulas, or the negations of these subformulas, may be added. Each branch is expanded until there are no variables left to make a binary split, or all the formulas of an abductive problem (or their negations) are added. No additional pruning is performed.

After the generation of the whole tableau, the failure branches are extracted. For each failure branch found, the set of the entangled literals is retrieved. Subsequent steps of the processing of the acquired literals and generation of the abducibles are described in Sect. 3.4.

3.2 Optimization issues

There are two computational bottlenecks in our framework. Firstly, the effective use of the synthetic rules is an issue: the optimal order of the application of the rules, or selection of the formulas to apply the rules to, are yet to be settled. For the time being, arbitrary choices and brute force approach are in effect. The other problem is that despite introduction of reduction, simplification and evaluation (Sect. 3.4 and 4.1), the set of abductive hypotheses might still be large since its size depends on the complexity of the considered problem. Although for the problems analyzed here reduction and simplification mechanisms work just fine, in general an overflow of the extensive amount of abducibles may be an issue.

A solution to these problems may be a heuristic or meta-heuristic approach. The optimization perspective includes a possibility to guide the search for interesting hypotheses using either some single parameter as their characteristics, or to employ a novel approach introduced in [21] consisting in considering multiple criteria simultaneously. The efficiency of heuristic search algorithms like multiple random start local search, Tabu search, simulated annealing, particle swarm, or evolutionary techniques [34, 12, 5] will be increased with the appropriate fitness landscape, so fitness-distance analyses are highly recommended [31, 15, 40].

3.3 The abductive problems considered and the problem generator

83 derivation problems have been considered, 14 of which have been created by a logician. The problems are numbered and denoted as Pxx in this paper, where xx is the number (the ID) of the problem. The problems discussed in more detail in the text are presented in the table in Sec. 4.2. Four of the expert-designed problems (P2, P9, P10, P11) were analyzed in earlier work [21]; the first three ones were also examined with a customizable filter denoted (A) in Sect. 3.4. In these three cases the results obtained were identical with and without the filter (A), and i.a. for this reason the filter is not used in this work, although the results could vary on a different sample of abductive problems. A new example of an expert-designed problem is P29; the majority (69) of abductive problems were however generated according to the following automated procedure:

- first, the set S containing four propositional variables and their negations was fixed, and then its power set 2^S was generated,
- each element of 2^S containing a pair of complementary literals was excluded,

- out of each two elements of 2^S of the same size of which one contained all the literals complementary to the literals in the second set, only one was left,
- from each of the remaining non-empty subsets X, Y of S , every possible formula of the form $\bigwedge X \rightarrow \bigvee Y$, where each literal might occur only once in the whole formula, was generated,
- then, a number of formulas were drawn from the set of all formulas generated in the previous step. Each formula had equal chance to be drawn, and for each formula the test whether to draw it or not was provided – thus the distribution of the number of the drawn formulas at one go was binominal. The selected formulas constituted the set of premises,
- the abductive goal was formed as a conjunction of two random, non-complementary literals each of which had to be present in at least one formula in the set of the premises,
- finally, a test whether A logically follows from X was conducted; pairs X, A which failed the test were considered abductive problems.

Following the above procedure it is unlikely (though still possible) to generate an inconsistent database. However, the entailment test conducted at the last step excludes the cases of inconsistent X . On the other hand, it is still possible to obtain a database X and a goal A such that $X \cup \{A\}$ is inconsistent (see P21, P55, P81) or such that $X \cup \{A\}$ is consistent but still some of the generated abducibles are inconsistent with X (like in the cases of P7, P18, P37, P58).

3.4 The set of abductive hypotheses and its reductions

For an abductive problem, the set of abductive hypotheses is generated by a procedure that goes through the following stages. First, a relevant synthetic tableau is generated (cf. (T1) in Section (2.2)), then:

1. The sets of entangled literals corresponding to the synthetic failures (as described in Sect. 2) are created – see (T2).
2. Subsequently, a number of reduction techniques are applied as described below. To illustrate this process, let us consider the following example:

$$\Delta_1 = \{\{p, q, r\}, \{p, q, \neg r\}, \{p, q\}, \{\neg p, q, s\}, \{\neg p, q, t\}\}$$

At this stage, each pair of the sets of literals – L_1 and L_2 – whose symmetric difference⁷ contained a pair of complementary literals (and only this pair), is replaced by their intersection: $L_i = L_1 \cap L_2$. This is done recursively until no such pair is left. After this step we are left with the following structure:

$$\Delta_2 = \{\{p, q\}, \{p, q\}, \{\neg p, q, s\}, \{\neg p, q, t\}\}$$

3. Then, all but one of the identical clauses of entangled literals are removed:

$$\Delta_3 = \{\{p, q\}, \{\neg p, q, s\}, \{\neg p, q, t\}\}$$

4. Next, the entangled literals are “complemented” (that is, for each literal its complement is considered). Each D_i is thus represented as a clause of literals, and the relevant formula E is represented as a family of clauses (see stage (T3)). In our example we arrive at:

$$\Delta_4 = \{\{\neg p, \neg q\}, \{p, \neg q, \neg s\}, \{p, \neg q, \neg t\}\}$$

The structure generated up to this point corresponds to a formula in conjunctive normal form (CNF).

⁷Symmetric difference is the union of two differences: $L_1 \ominus L_2 = (L_1 \setminus L_2) \cup (L_2 \setminus L_1)$.

5. The family of the remaining clauses of entangled literals is transformed into a formula in disjunctive normal form (DNF, this stage corresponds to (T4)). Thus each clause now represents a conjunctive clause. Then all repetitions of literals in clauses are removed. This yields the following formula (in clausal representation):

$$\Delta_5 = \{\{\neg p, p\}, \{\neg p, p, \neg q\}, \{\neg p, p, \neg t\}, \{\neg p, \neg q, p\}, \{\neg p, \neg q\}, \{\neg p, \neg q, \neg t\}, \\ \{\neg p, \neg s, p\}, \{\neg p, \neg s, \neg q\}, \{\neg p, \neg s, \neg t\}, \{\neg q, p\}, \{\neg q, p\}, \{\neg q, p, \neg t\}, \{\neg q, p\}, \\ \{\neg q\}, \{\neg q, \neg t\}, \{\neg q, \neg s, p\}, \{\neg q, \neg s\}, \{\neg q, \neg s, \neg t\}\}$$

6. All the conjunctive clauses containing at least one pair of complementary literals are removed (subsequently, inconsistent sets are filtered out):

$$\Delta_6 = \{\{\neg p, \neg q\}, \{\neg p, \neg q, \neg t\}, \{\neg p, \neg s, \neg q\}, \{\neg p, \neg s, \neg t\}, \{\neg q, p\}, \{\neg q, p\}, \\ \{\neg q, p, \neg t\}, \{\neg q, p\}, \{\neg q\}, \{\neg q, \neg t\}, \{\neg q, \neg s, p\}, \{\neg q, \neg s\}, \{\neg q, \neg s, \neg t\}\}$$

7. All but one of the identical clauses are cleared out:

$$\Delta_7 = \{\{\neg p, \neg q\}, \{\neg p, \neg q, \neg t\}, \{\neg p, \neg s, \neg q\}, \{\neg p, \neg s, \neg t\}, \{\neg q, p\}, \{\neg q, p, \neg t\}, \\ \{\neg q\}, \{\neg q, \neg t\}, \{\neg q, \neg s, p\}, \{\neg q, \neg s\}, \{\neg q, \neg s, \neg t\}\}$$

8. Then, steps (2) and (3) are once again applied to the clauses of literals:

$$\Delta_8 = \{\{\neg q\}, \{\neg q, \neg t\}, \{\neg q, \neg s\}, \{\neg p, \neg s, \neg t\}, \{\neg q, \neg s, \neg t\}\}$$

9. All conjunctive clauses represented by sets of literals that are strict supersets of some other sets of literals are removed:

$$\Delta_9 = \{\{\neg q\}, \{\neg p, \neg s, \neg t\}\}$$

- (A) In an additional reduction stage (A), for the problems where the abductive goal consisted of a single literal, all the conjunctive clauses containing this literal can be cleared out. Assuming that in our example the abductive goal is $\neg q$, we may also clear out the subset containing this literal, and thus our only abductive hypothesis would be $(\neg p \wedge \neg s) \wedge \neg t$.
- (10) After these preliminary reductions we arrive at a set of conjunctive clauses. Each non-empty subset of this set (including singletons and the set itself) is then expressed as a formula in DNF. The output of this process is the set of abductive hypotheses.
- (11) For this set, the final simplifying reduction has been performed using the Quine-McCluskey algorithm.

The Quine-McCluskey algorithm [27, 17] is also referred to as the method of prime implicants. The algorithm, sometimes called the tabulation method, is a more efficient version of the ‘‘Karnaugh mapping’’ [41, 19]. This procedure has been applied to each hypothesis to simplify it, which allowed to identify subsets of hypotheses logically equivalent on the grounds of CPC. This final reduction step resulted in further decrease of the number of hypotheses, which – after this step – were guaranteed to be semantically unique.

Introducing all the reduction techniques described above is a major improvement compared to our previous work [21], where no such methods were used, and our considerations were based on hypotheses resulting from a procedure going through stages (1),(4),(5) and (10) – called here the ‘basic procedure’. Stages (2),(3) and (6)-(9), which are novel in this work, are of purely reductive character. Their aim is to reduce the space of abducibles by eliminating, first, ‘‘repetitions’’ (that is, logically equivalent but syntactically distinct formulas), and second, hypotheses which are logically too strong in that they entail

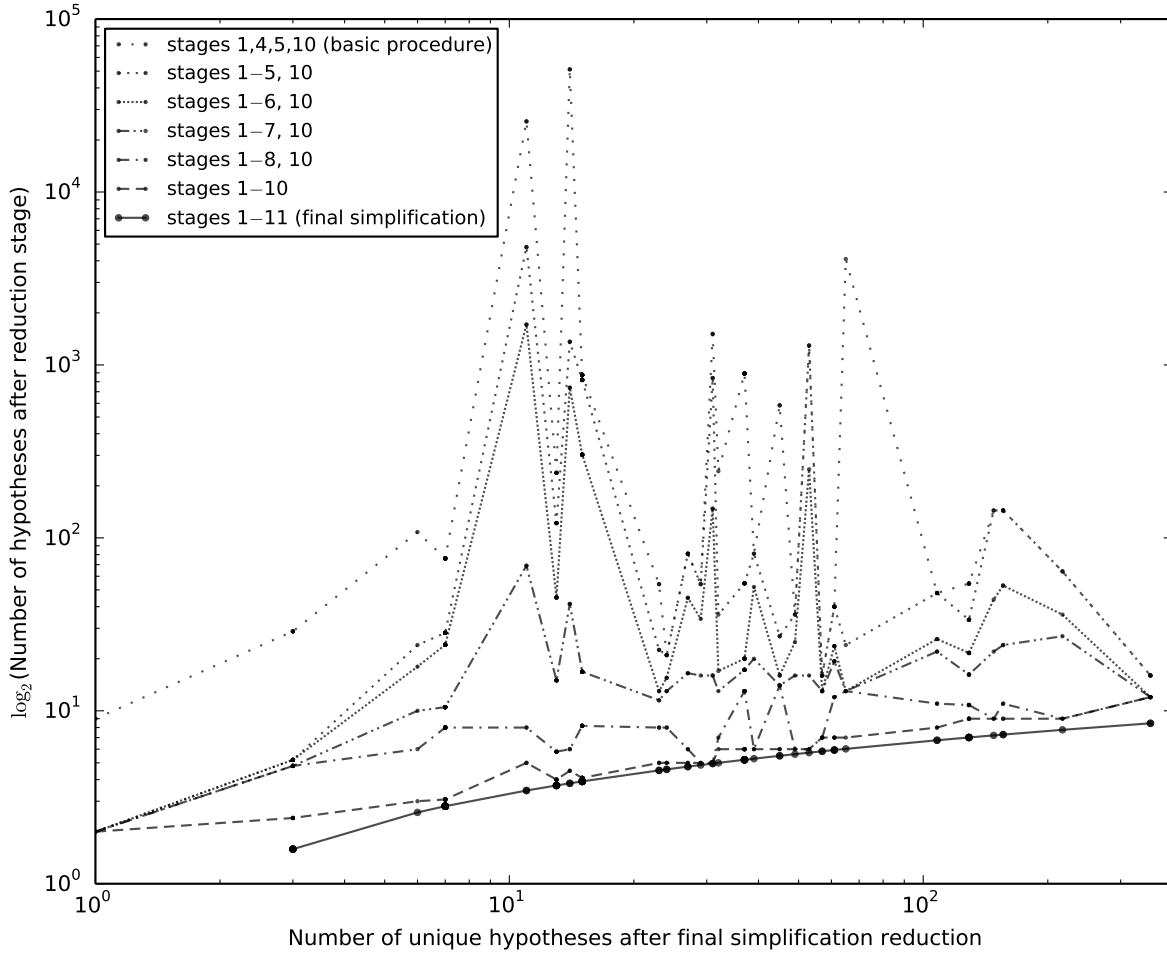


Figure 1: Efficiency of the reduction stages described in the text. Note that the vertical axis is doubly logarithmic to include sizes of non-reduced sets of hypotheses; the solid line is provided for reference as $y = x$. To increase readability of the plot, where multiple problems yielded the same number of hypotheses after final reduction (the same value on the x axis), the number of hypotheses after non-final reductions (values on the y axis) have been averaged so that such problems constitute one point on the plot.

some other abducibles which, at the same time, are syntactically less complex (this is the goal of stage (9)).

The efficiency of the reduction stages described above is demonstrated in Fig. 1. Label ‘basic procedure’ refers to the set of hypotheses obtained by the original procedure as in [21], ‘stages 1-5,10’ refers to the set of hypotheses generated according to stages (1)-(5) and (10) (thus reduction steps (2) and (3) are included), etc. ‘Final simplification’ refers to the outcomes of the whole procedure (1)-(11) with all reduction stages employed. Note that the vertical axis covers a huge range of numbers of hypotheses because it is doubly logarithmic; this illustrates enormous reductions that take place in each reduction stage.⁸ The graph shows that the amount of reduction provided by each stage differs

⁸Let us describe one particular example here. In the case of problem P29 (see table below) the database and the abductive goal are expressed by the use of five distinct variables. There are exactly 2^{2^5} different truth-functions that may

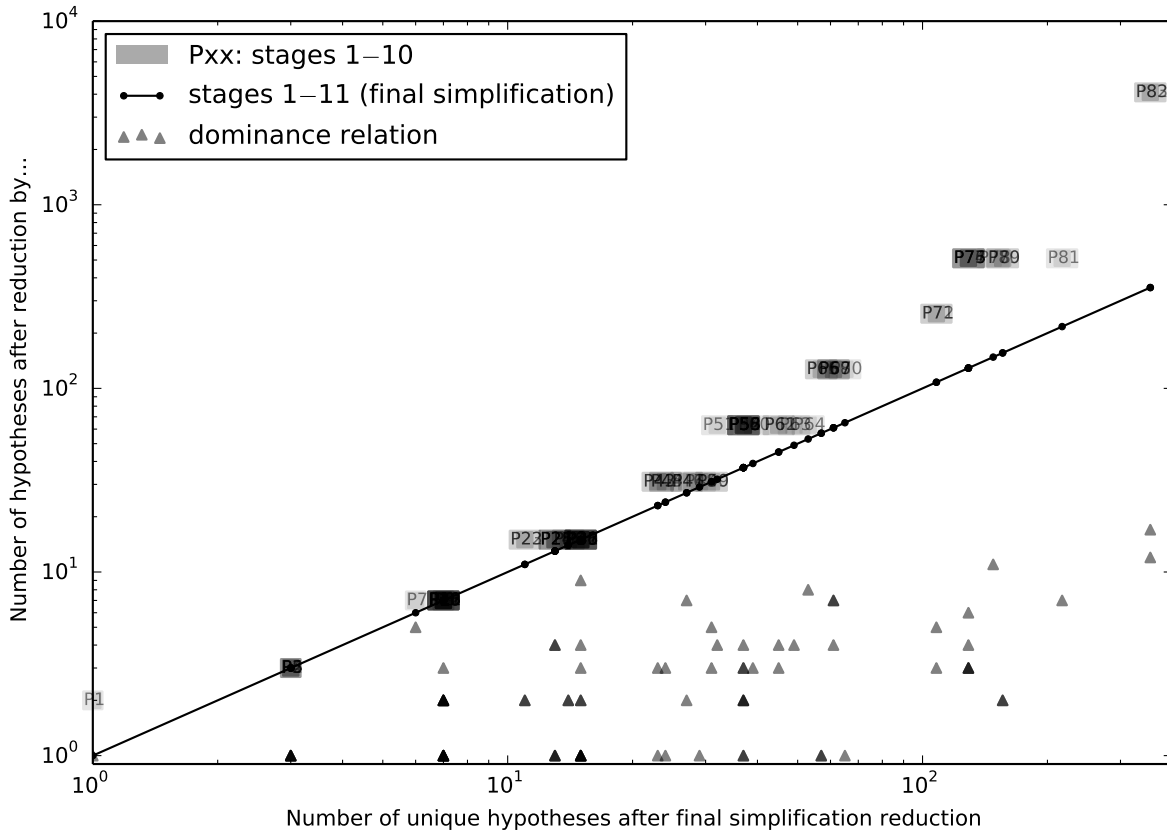


Figure 2: Efficiency of the final reduction stage (shown as a reference: solid line, $y = x$) compared to preliminary reductions (stages 1-10, Pxx labels show IDs of individual problems) and reductions yielded by the application of the dominance relation using three evaluation criteria (triangles correspond to labels in Fig. 3).

highly from problem to problem. While there is no absolute dependence between the final number of unique hypotheses and the number of hypotheses obtained from earlier reduction stages (the lines are not monotonically increasing), one can notice some correlation as a general increasing trend is visible. The raw number of hypotheses is the poorest predictor of the final number of unique hypotheses. The characteristics of the last stage (1-10) before the ultimate simplification process is more clearly shown in Fig. 2, where the vertical axis is singly logarithmic, just as the horizontal axis. This figure additionally shows reduction of the set of hypotheses provided by the dominance relation described in the following section.

be expressed by formulas formed in the standard language of CPC with five distinct variables. And yet we come across 2^{102400} abductive hypotheses (formulated with only the five variables) generated by the basic procedure. When the space of abducibles is generated according to the pattern described in Section 3.4, we get 15 hypothesis after the preliminary reductions and after the final stage (11) we are left with 14 of them.

Observe, however, that the reduction mechanism applied in this work is strongly logic dependent, thus it would produce incorrect results were we to depart the safe ground of CPC. We find it interesting to generate abductive hypotheses by means of a non-classically designed proof procedure and/or to evaluate abductive hypotheses with non-classically motivated criteria. In these cases the results reported in the previous work [21] would be a more appropriate starting point.

4 Identifying interesting hypotheses using multi-criteria evaluation

4.1 Evaluation criteria

There exist several criteria meant to evaluate abductive hypotheses [1]. Here we will briefly comment on how they are implemented. While the STM might be used to perform calculations important to establish values of the criteria, to speed up the evaluation process we decided to implement the Analytic Tableaux Method (ATM) and use it as a main proof method. The increase in speed in our case is due to the higher algorithmic explicitness of the ATM; as we have already observed, there are no good heuristics guiding the priorities of rules applications in STM. To speed up the process even more, the syntactic analysis of the branches of analytic tableaux is done on the fly (not after generation of the whole tableau), which also avails the pre-pruning techniques. Each criterion is implemented as a function which takes some arguments and returns an integer value; arrows indicate preference (\uparrow the more the better, \downarrow the less the better).

Recall that X represents a database, A represents an abductive goal and H an abductive hypothesis (abducible). Let us also recall for clarity that an analytic tableau for a formula F is a tableau with formula $\neg F$ in the root.

1. \uparrow Consistency – this function builds an analytic tableau for the formula $\bigwedge X \rightarrow \neg H$. If the tableau is closed, the function returns 0, otherwise it returns 1. The idea behind this criterion is that with CPC as the basic logic, H should be consistent with X .
2. \uparrow Significance – the function builds an analytic tableau for the formula $H \rightarrow A$. If the tableau is closed, the function returns 0, which means that H itself entails A and the set of premises (the database) is not significant; otherwise the function returns 1. (Thus the criterion measures the ‘significance’ of the database relative to the hypothesis.)
3. \downarrow Complexity – this function takes one argument (the formula H) and returns the number of distinct variables found in the formula.
4. \downarrow Operational Complexity – this function takes one argument (the formula H) and returns the number of two-argument operators used to build the formula (that is, the number of occurrences of \wedge and \vee ; recall that H after the final simplification stage is expressed with \neg , \wedge and \vee only). The two complexities are somewhat converging, nevertheless in some cases they yield different results – consider for example “fat” expressions, which are lengthy with a relatively small number of distinct variables [8].
5. \downarrow Overlapping – this function takes two arguments (the formula A and the hypothesis H) and returns the number of variables each of which occurs in both formulas.⁹

After the evaluation of a hypothesis using each of the criteria described above, the criteria values may be aggregated or not, depending on the research goals. In this work we avoid aggregation and thus avoid introducing trade-offs between these criteria. Since they are often conflicting, we use the multi-criteria dominance relation to identify hypotheses that are better than others in any aspect, and to capture a human goal of finding *interesting* abductive hypotheses. Formally, we assume that hypothesis H_1 is better than H_2 (i.e., dominates it), when H_1 is not worse (which means it is better or equally good) than H_2 on all criteria, and H_1 is strictly better than H_2 on at least one of these criteria. A hypothesis H that is not dominated by any other hypotheses in the set is called a non-dominated hypothesis, an efficient hypothesis, or a Pareto optimal hypothesis [9, 7].

⁹This criterion was called “Relevance” in the previous work [21]. The idea was to maintain the connection of contents of H and A , thus Relevance was maximized. However, maximized Relevance mainly favoured hypotheses syntactically identical with abductive goals. For this reason the name of the criterion as well as its preference were changed.

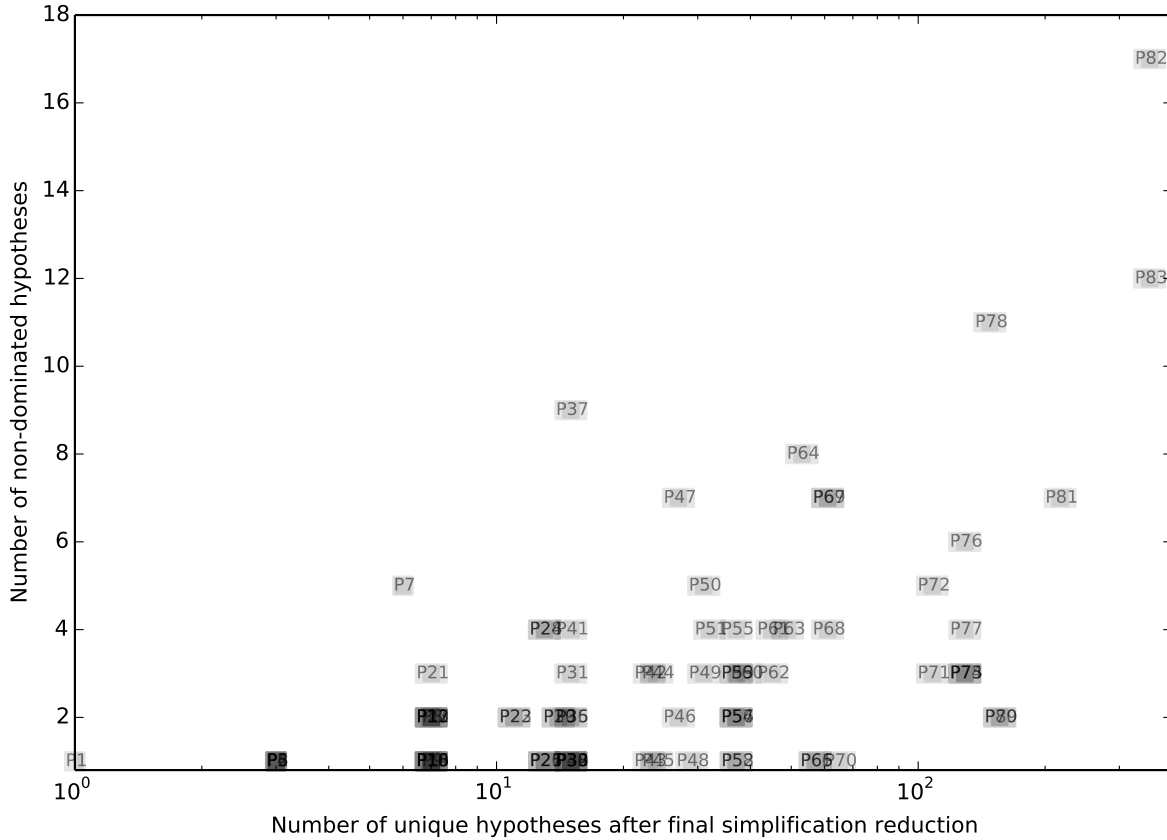


Figure 3: Efficiency of reduction by the dominance relation: removal of the dominated hypotheses according to their 3-criteria evaluation (C3). Note that the vertical axis is linear while the horizontal axis remains logarithmic.

Note an interesting characteristic of the dominance relation defined above: introducing more criteria will likely increase the number of non-dominated hypotheses as long as the criteria are in conflict (i.e., when comparing two hypotheses, one hypothesis is better according to some criterion and worse according to some other criterion). However, adding new criteria may also reduce the set of non-dominated hypotheses, as new criteria may provide additional information that will differentiate hypotheses which were previously identically evaluated. In any case, the dominance relation allows one to reduce the set of hypotheses and thus to obtain a smaller set of interesting ones without the need of introducing strong, arbitrary assumptions (like imposing some order or hierarchy of criteria) and aggregation models (like a weighted sum of criteria values).

For each problem, each pair of hypotheses in the set of unique hypotheses that resulted from the final simplification reduction was compared using the multi-criteria dominance relation in order to find subsets of non-dominated hypotheses. This allowed for further reduction of the number of hypotheses, as Fig. 3 (and Fig. 2) illustrates. For most problems considered here, the set of non-dominated hypotheses was small enough to be analyzed by a human expert, but for larger problems, the dominance relation may be too weak to significantly reduce the set. The trend is however encouraging, and seems quasi-linear with exponentially increasing size of the set of unique hypotheses.

4.2 Discussion of the outcomes of the reduction based on multi-criteria dominance

The table below summarizes results of multi-criteria evaluation for problems that are discussed in the text. For each problem, non-dominated hypotheses are shown for the following sets of criteria:

- (C3) Consistency, Significance, Complexity
- (C4a) Consistency, Significance, Complexity, Overlapping
- (C4b) Consistency, Significance, Complexity, Operational Complexity
- (C5) Consistency, Significance, Complexity, Overlapping, Operational Complexity.

Problem ID, number of variables, size of the set of hypotheses: basic procedure (stages 1,4,5,10), preliminary reductions (stages 1–10), final simplification (stages 1–11)	set of formulas ("database")	abductive goal	non-dom count for: C3 C4a C4b C5	criteria values	non-dominated hypotheses
P2 variables: 3 $2^9, 2^2 - 1 = 3, 3$	$(p \wedge r) \rightarrow q$ $(\neg r \rightarrow q) \rightarrow p$	q	1 1 1 1	1,1,1 1,1,1,0 1,1,1,0 1,1,1,0,0	r r r r
P7 variables: 4 $2^{108}, 2^3 - 1 = 7, 6$	$(p \wedge q) \rightarrow (s \vee \neg r)$ $p \rightarrow (q \vee \neg r)$ $(p \wedge \neg q) \rightarrow r$ $q \rightarrow (\neg r \vee \neg s)$	$r \wedge q$	5 5 4 4	1,1,3 0,1,2 1,1,3 1,0,2 0,1,2 1,1,3,2 0,1,2,1 1,1,3,2 1,0,2,2 0,1,2,1 0,1,2,1 1,1,3,2 1,0,2,1 0,1,2,1 0,1,2,1,1 1,1,3,2,2 1,0,2,2,1 0,1,2,1,1	$(p \wedge \neg q) \vee (q \wedge r)$ $p \wedge r$ $r \wedge (p \vee q)$ $q \wedge r$ $p \wedge \neg q$ $(p \wedge \neg q) \vee (q \wedge r)$ $p \wedge r$ $r \wedge (p \vee q)$ $q \wedge r$ $p \wedge \neg q$ $p \wedge r$ $r \wedge (p \vee q)$ $q \wedge r$ $p \wedge \neg q$ $p \wedge r$ $r \wedge (p \vee q)$ $q \wedge r$ $p \wedge \neg q$
P11 variables: 5 $2^{12}, 2^3 - 1 = 7, 7$	$(\neg r \vee q) \rightarrow s$ $\neg t \rightarrow (p \wedge q)$ $q \rightarrow \neg r$	$q \wedge s$	2 1 2 1	1,1,1 1,1,1 1,1,1,0 1,1,1,0 1,1,1,0,0	q $\neg t$ $\neg t$ q $\neg t$ $\neg t$
P16 variables: 4 $2^{54}, 2^3 - 1 = 7, 7$	$(r \wedge \neg s) \rightarrow (p \vee q)$ $(p \wedge q) \rightarrow (\neg r \vee \neg s)$ $((p \wedge q) \wedge \neg s) \rightarrow \neg r$ $((p \wedge r) \wedge \neg q) \rightarrow s$ $(r \wedge \neg s) \rightarrow (p \vee \neg q)$ $p \rightarrow (r \vee \neg s)$ $(s \wedge \neg q) \rightarrow (p \vee \neg r)$ $((p \wedge s) \wedge \neg r) \rightarrow \neg q$ $p \rightarrow \neg r$	$s \wedge q$	1 1 1 1	1,1,1 1,1,1,0 1,1,1,0 1,1,1,0,0	r r r r
P17 variables: 4 $2^{16}, 2^3 - 1 = 7, 7$	$r \rightarrow ((p \vee q) \vee \neg s)$ $p \rightarrow (q \vee r)$ $(q \wedge \neg r) \rightarrow p$ $(r \wedge \neg q) \rightarrow (p \vee \neg s)$ $\neg q \rightarrow (p \vee s)$ $(p \wedge \neg r) \rightarrow \neg q$ $\neg q \rightarrow (p \vee \neg s)$ $(p \wedge \neg q) \rightarrow \neg s$ $(q \wedge \neg s) \rightarrow r$	$\neg s \wedge r$	2 1 2 1	1,1,1 1,1,1 1,1,1,0 1,1,1,0 1,1,1,0,0	$\neg q$ $\neg s$ $\neg q$ $\neg q$ $\neg s$ $\neg s$ $\neg q$ $\neg q$
P18 variables: 4 $2^{108}, 2^3 - 1 = 7, 7$	$((p \wedge r) \wedge s) \rightarrow q$ $((p \wedge q) \wedge s) \rightarrow r$ $q \rightarrow (p \vee s)$ $p \rightarrow ((r \vee \neg q) \vee \neg s)$ $((p \wedge r) \wedge \neg s) \rightarrow \neg q$ $((p \wedge s) \wedge \neg q) \rightarrow \neg r$ $q \rightarrow (r \vee s)$	$p \wedge r$	1 2 1 2	1,1,2 1,1,2,1 0,1,2,0 1,1,2,1 1,1,2,1,1 0,1,2,0,1	$p \wedge q$ $p \wedge q$ $q \wedge \neg s$ $p \wedge q$ $p \wedge q$ $q \wedge \neg s$
P21 variables: 4 $2^{12}, 2^3 - 1 = 7, 7$	$s \rightarrow ((p \vee q) \vee r)$ $(q \wedge s) \rightarrow (p \vee r)$ $((p \wedge q) \wedge \neg s) \rightarrow r$ $q \rightarrow p$ $((p \wedge s) \wedge \neg q) \rightarrow r$ $(p \wedge r) \rightarrow (s \vee \neg q)$ $((p \wedge r) \wedge \neg q) \rightarrow s$ $s \rightarrow (p \vee \neg q)$ $(\neg q \wedge \neg s) \rightarrow p$ $q \rightarrow (s \vee \neg r)$ $\neg r \rightarrow q$ $q \rightarrow \neg s$	$\neg s \wedge p$	3 2 3 2	0,1,1 0,1,1 0,1,1 0,1,1,0 0,1,1,0 0,1,1,0 0,1,1,0 0,1,1,0 0,1,1,0,0 0,1,1,0,0	q $\neg r$ $\neg s$ q $\neg r$ q $\neg r$ $\neg s$ $\neg s$ q $\neg r$
P29 variables: 5 $2^{102400}, 2^4 - 1 = 15, 14$	$p \rightarrow (q \vee r)$ $\neg t \rightarrow \neg r$ $q \rightarrow (s \vee t)$ $s \rightarrow (\neg t \vee r)$	s	2 2 2 2	1,1,2 1,1,2 1,1,2,0 1,1,2,0 1,1,2,1 1,1,2,1 1,1,2,0,1 1,1,2,0,1	$p \wedge \neg t$ $q \wedge \neg t$ $p \wedge \neg t$ $q \wedge \neg t$ $p \wedge \neg t$ $q \wedge \neg t$ $p \wedge \neg t$ $q \wedge \neg t$
P37 variables: 4 $2^{243}, 2^4 - 1 = 15, 15$	$r \rightarrow ((p \vee q) \vee s)$ $(p \wedge q) \rightarrow (s \vee \neg r)$ $q \rightarrow (p \vee s)$ $(p \wedge q) \rightarrow (\neg r \vee \neg s)$ $p \rightarrow \neg s$	$q \wedge r$	9	1,1,4 1,1,4 1,1,4 1,1,4 1,1,4 1,1,4 1,1,4 1,1,4 1,1,4 1,1,4	$r \wedge (q \vee \neg p) \wedge (q \vee \neg s)$ $(p \vee q) \wedge (p \vee r) \wedge (q \vee s) \wedge (r \vee s)$ $(p \wedge s) \vee (q \wedge r) \vee (r \wedge \neg p \wedge \neg s)$ $q \wedge (r \vee \neg p) \wedge (r \vee \neg s)$ $p \wedge s$ $q \wedge r$ $(q \vee r) \wedge (q \vee \neg p) \wedge (q \vee \neg s) \wedge (r \vee \neg p) \wedge (r \vee \neg s)$ $(p \wedge s) \vee (q \wedge r) \vee (q \wedge \neg p \wedge \neg s)$ $(p \wedge s) \vee (q \wedge r) \vee (q \wedge \neg p \wedge \neg s) \vee (r \wedge \neg p \wedge \neg s)$

Problem ID, number of variables, size of the set of hypotheses: basic procedure (stages 1,4,5,10), preliminary reductions (stages 1–10), final simplification (stages 1–11)	set of formulas ("database")	abductive goal	non-dom count for: C3 C4a C4b C5	criteria values	non-dominated hypotheses
			9	1,1,4,2 1,1,4,2 1,1,4,2 1,1,4,2 0,1,2,0 1,0,2,2 1,1,4,2 1,1,4,2 1,1,4,2	$r \wedge (q \vee \neg p) \wedge (q \vee \neg s)$ $(p \vee q) \wedge (p \vee r) \wedge (q \vee s) \wedge (r \vee s)$ $(p \wedge s) \vee (q \wedge r) \vee (r \wedge \neg p \wedge \neg s)$ $q \wedge (r \vee \neg p) \wedge (r \vee \neg s)$ $p \wedge s$ $q \wedge r$ $(q \vee r) \wedge (q \vee \neg p) \wedge (q \vee \neg s) \wedge (r \vee \neg p) \wedge (r \vee \neg s)$ $(p \wedge s) \vee (q \wedge r) \vee (q \wedge \neg p \wedge \neg s)$ $(p \wedge s) \vee (q \wedge r) \vee (q \wedge \neg p \wedge \neg s) \vee (r \wedge \neg p \wedge \neg s)$
			4	1,1,4,4 1,1,4,4 0,1,2,1 1,0,2,1	$r \wedge (q \vee \neg p) \wedge (q \vee \neg s)$ $q \wedge (r \vee \neg p) \wedge (r \vee \neg s)$ $p \wedge s$ $q \wedge r$
			4	1,1,4,2,4 1,1,4,2,4 0,1,2,0,1 1,0,2,2,1	$r \wedge (q \vee \neg p) \wedge (q \vee \neg s)$ $q \wedge (r \vee \neg p) \wedge (r \vee \neg s)$ $p \wedge s$ $q \wedge r$
P55 variables: 4 $2^{10} 2^4, 2^6 - 1 = 63, 37$	$(p \wedge q) \rightarrow r$ $(q \wedge s) \rightarrow (p \vee \neg r)$ $(p \wedge \neg s) \rightarrow (q \vee \neg r)$ $((p \wedge r) \wedge s) \rightarrow \neg q$ $(r \wedge \neg q) \rightarrow p$ $s \rightarrow ((p \vee \neg q) \vee \neg r)$ $(\neg r \wedge \neg s) \rightarrow p$ $\neg s \rightarrow (q \vee r)$ $q \rightarrow r$ $\neg s \rightarrow q$	$\neg s \wedge \neg r$	4	0,1,2 0,1,2 0,1,2 0,1,2	$q \wedge s$ $q \wedge \neg r$ $\neg q \wedge \neg s$ $(q \wedge s) \vee (\neg q \wedge \neg s)$
			4	0,1,2,1 0,1,2,1 0,1,2,1 0,1,2,1	$q \wedge s$ $q \wedge \neg r$ $\neg q \wedge \neg s$ $(q \wedge s) \vee (\neg q \wedge \neg s)$
			3	0,1,2,1 0,1,2,1 0,1,2,1	$q \wedge s$ $q \wedge \neg r$ $\neg q \wedge \neg s$
			3	0,1,2,1,1 0,1,2,1,1 0,1,2,1,1	$q \wedge s$ $q \wedge \neg r$ $\neg q \wedge \neg s$
P58 variables: 4 $2^{27}, 2^6 - 1 = 63, 37$	$((p \wedge q) \wedge s) \rightarrow r$ $q \rightarrow (p \vee r)$ $p \rightarrow (q \vee \neg r)$ $s \rightarrow (p \vee r)$ $r \rightarrow (p \vee s)$ $(p \wedge \neg q) \rightarrow (s \vee \neg r)$ $(\neg q \wedge \neg r) \rightarrow p$ $\neg s \rightarrow p$ $s \rightarrow (q \vee r)$	$\neg s \wedge r$	1	1,1,2	$(r \wedge \neg s) \vee (s \wedge \neg r)$
			3	1,1,2,2 0,1,2,1 0,1,2,1	$(r \wedge \neg s) \vee (s \wedge \neg r)$ $\neg p \wedge \neg s$ $\neg p \wedge \neg r$
			6	1,1,3,2 0,1,2,1 1,1,2,3 0,1,2,1 0,1,2,1 1,0,2,1	$\neg s \wedge (r \vee \neg p)$ $s \wedge \neg r$ $(r \wedge \neg s) \vee (s \wedge \neg r)$ $\neg p \wedge \neg s$ $\neg p \wedge \neg r$ $r \wedge \neg s$
			5	1,1,3,2,2 1,1,2,2,3 0,1,2,1,1 0,1,2,1,1 1,0,2,2,1	$\neg s \wedge (r \vee \neg p)$ $(r \wedge \neg s) \vee (s \wedge \neg r)$ $\neg p \wedge \neg s$ $\neg p \wedge \neg r$ $r \wedge \neg s$
P66 variables: 4 $2^{16}, 2^7 - 1 = 127, 57$	$q \rightarrow p$ $((p \wedge r) \wedge \neg q) \rightarrow s$ $((p \wedge r) \wedge s) \rightarrow \neg q$ $p \rightarrow (r \vee \neg s)$ $(\neg q \wedge \neg r) \rightarrow p$ $\neg s \rightarrow (q \vee r)$ $q \rightarrow (s \vee \neg r)$	$r \wedge s$	1	1,1,1	s
			3	1,1,2,0 1,1,1,1 1,1,2,0	$(p \wedge \neg q) \vee (q \wedge \neg p)$ s $p \wedge \neg q$
			1	1,1,1,0	s
			2	1,1,1,1,0 1,1,2,0,1	s $p \wedge \neg q$
P80 variables: 4 $2^{144}, 2^9 - 1 = 511, 156$	$((p \wedge \neg r) \wedge \neg s) \rightarrow q$ $\neg q \rightarrow ((p \vee r) \vee \neg s)$ $r \rightarrow (p \vee \neg s)$ $p \rightarrow (s \vee \neg r)$ $(\neg r \wedge \neg s) \rightarrow p$ $(q \wedge r) \rightarrow s$ $(\neg r \wedge \neg s) \rightarrow q$	$s \wedge \neg q$	2	1,1,2 1,1,2	$p \wedge \neg q$ $\neg q \wedge \neg r$
			2	1,1,2,1 1,1,2,1	$p \wedge \neg q$ $\neg q \wedge \neg r$
			2	1,1,2,1 1,1,2,1	$p \wedge \neg q$ $\neg q \wedge \neg r$
			2	1,1,2,1,1 1,1,2,1,1	$p \wedge \neg q$ $\neg q \wedge \neg r$

Problem ID, number of variables, size of the set of hypotheses: basic procedure (stages 1,4,5,10), preliminary reductions (stages 1–10), final simplification (stages 1–11)	set of formulas ("database")	abductive goal	non-dom count for: C3 C4a C4b C5	criteria values	non-dominated hypotheses
P81 variables: 4 $2^{64}, 2^9 - 1 = 511, 217$	$q \rightarrow ((p \vee r) \vee s)$ $(p \wedge r) \rightarrow (q \vee s)$ $(p \wedge \neg r) \rightarrow (q \vee s)$ $((p \wedge \neg r) \wedge \neg s) \rightarrow q$ $(q \wedge \neg r) \rightarrow p$ $s \rightarrow ((p \vee \neg q) \vee \neg r)$ $((s \wedge \neg q) \wedge \neg r) \rightarrow p$ $((p \wedge \neg q) \wedge \neg r) \rightarrow s$ $(q \wedge s) \rightarrow r$ $r \rightarrow (q \vee \neg s)$ $q \rightarrow s$ $\neg s \rightarrow r$	$\neg s \wedge q$	7	0,1,2 0,1,2 0,1,2 0,1,2 0,1,2 0,1,2 0,1,2	$q \wedge \neg r$ $(p \wedge \neg s) \vee (s \wedge \neg p)$ $q \wedge \neg p$ $\neg p \wedge \neg r$ $p \wedge \neg s$ $s \wedge \neg p$ $\neg r \wedge \neg s$
			1	0,1,2,0	$\neg p \wedge \neg r$
			6	0,1,2,1 0,1,2,1 0,1,2,1 0,1,2,1 0,1,2,1 0,1,2,1	$q \wedge \neg r$ $q \wedge \neg p$ $\neg p \wedge \neg r$ $p \wedge \neg s$ $s \wedge \neg p$ $\neg r \wedge \neg s$
			1	0,1,2,0,1	$\neg p \wedge \neg r$

Due to the fact that the reduction mechanisms together with the multi-criteria dominance have usually yielded a small number of abducibles, it is easy to verify whether we have arrived at desirable results – that is, whether the resulting abductive hypotheses are really non-trivial fillers of the deductive gaps. Let us consider results of a few selected problems.

In the case of problem P16, no matter the set of criteria, the only non-dominated hypothesis out of seven left after the reduction is r . It can be checked that this formula is consistent with the set of premises. It is also easily seen that it is significant (i.e. it does not entail $s \wedge q$ on its own), it does not “overlap” with the abductive goal and it is as simple as possible in terms of syntactic complexity. A closer look at the set of premises reveals that adding r to the 1st, 3rd, 4th and 5th premise yields s (since assuming r and $\neg s$ we arrive at a contradiction). From r and the last premise we have $\neg p$, and finally, from $r, s, \neg p$ and the 7th premise we derive q .

A more complex example is problem P17. Here, depending on the chosen set of criteria we are either left with formulas $\neg q$ and $\neg s$, or with the sole formula $\neg q$ out of the seven abducibles remaining after the reduction process. One can see that it is the overlapping criterion which causes formula $\neg s$ to be dominated by formula $\neg q$ (that is, hypothesis $\neg s$ occurs to be a worse solution than $\neg q$ since it constitutes half of the problem to be solved). Note that here the inclusion of the additional criterion resulted in diminishing the number of non-dominated hypotheses, which is not the case in general, as we will see in the next example. As in the previous case, it may be easily observed that the abducible $\neg q$ meets the criteria of consistency, significance, overlapping, and complexity. As for the derivative properties of this hypothesis, we can see that from $\neg q$, the 5th and the 7th premises we arrive at p ; this, together with the 8th premise, gives us $\neg s$; and finally, from $p, \neg q$ and the 2nd premise we have r .

Some interesting issues connected with the dominance relation are demonstrated by problem P66; one can see that taking into consideration just the three criteria of consistency, significance and complexity results in only one non-dominated abducible, s , out of the 57 simplified abductive hypotheses. Hypothesis s is not very satisfactory because it overlaps with the abductive goal. This situation is not changed by the inclusion of the operational complexity criterion. However, when instead of the operational complexity criterion the overlapping criterion is included, two additional abducibles appear to be non-dominated: $(p \wedge \neg q) \vee (q \wedge \neg p)$ and $p \wedge \neg q$. The inclusion of all five of the criteria leaves us with two abducibles: s and $p \wedge \neg q$. Intuitively, the most interesting abducible seems to be $p \wedge \neg q$, which is not as simple as s , but does not overlap with the abductive goal. Observe that the set of three “basic” criteria of consistency, significance and complexity is not sensitive enough to capture this intuition. Similarly, in the case of P18, if we include three criteria, then $p \wedge q$ is the only non-dominated hypothesis, but it partially overlaps with the abductive goal, and for this reason $q \wedge \neg s$ is also included when we add the criterion of overlapping. These examples illustrate how sensitive the dominance relation might be to the set of criteria and that, in general, the set of the non-dominated abducibles does not change monotonically with the increasing number of criteria used to evaluate the hypotheses.

This situation seems similar in problem P58, where the only non-dominated hypothesis obtained under the three criteria is definitely of little value. Note that the first argument of disjunction is equivalent to the abductive goal and the second is inconsistent with the set of premises. Thus the hypothesis is significant (does not entail the abductive goal) only because a formula inconsistent with the database is added as the second argument of disjunction, and it is consistent with the set of premises only because a non-significant formula is added as the first. A more sophisticated set of criteria is needed to obtain other hypotheses.

Example P55 illustrates the case where the abductive goal is inconsistent with the set of premises.¹⁰ In fact, it is inconsistent with the last two premises. The inconsistency is less obvious in the case of P21. As one can see, in both cases the non-dominated hypotheses are always inconsistent with the set of premises, which must be the case with CPC as basic logic. But still, even among the hypotheses which do not conform the consistency criterion, the dominance relation allows to identify hypotheses which are interesting from the viewpoint of the remaining criteria. Thus the examples illustrate the fact that multi-criteria dominance relation may be profitably used with paraconsistent logics in the background. It should be noted that such a change of basic logic would require adjustment both in STM implementation (cf. [38]) and in hypotheses generation procedure.

In almost all of the problems considered here, the set of criteria denoted as (C4b) brings as non-dominated the hypotheses which were also provided by other sets of criteria. Problem P58 with hypothesis $s \wedge \neg r$ is an exception here. This lack of novelties for (C4b) is due to the fact that the two criteria of complexity are mostly concordant. The criterion of Operational Complexity added to the three “basic” criteria could probably yield more varied results for more complex formulas.

4.3 Discordance of criteria

It is possible to estimate how often each pair of criteria disagrees, i.e., indicates opposite preference for a pair of hypotheses (one criterion indicates that hypothesis H_1 is better than H_2 , while the other criterion indicates that H_2 is better than H_1). To estimate discordance of each pair of criteria for a given problem, all $n(n-1)/2$ pairs of n hypotheses are compared and opposite preferences are counted. Note that the situation where one criterion does not distinguish between two hypotheses and the other criterion prefers one hypotheses over the other is not considered a conflict.

Results shown in Fig. 4 demonstrate that discordance is usually low (especially among Significance, Overlapping, and the two Complexities) which explains why there are so few non-dominated hypotheses. In most cases Consistency is the most conflicting criterion – which means that it is hard to find a consistent hypothesis that is at the same time relatively good at other criteria. The high discordance between Consistency and Overlapping or Complexities (an example is problem P80) is caused by relatively frequent occurrences of hypotheses inconsistent with databases – which is probably a cue that the reduction process can be still improved by restricting hypotheses to be consistent with their databases (on the other hand, this situation may be somehow exploited with non-classical logics, as we have already observed). When the abductive goal is inconsistent with the database, Consistency is in total agreement with the other criteria; examples of such problems are P81 and P82. Therefore, all of the abductive hypotheses for these problems have to be inconsistent with the sets of premises. This situation entails maximal agreement between Consistency and the other criteria.

The discordance between Significance and Complexities is rather low partially due to the fact that after the reduction processes there are few non-significant hypotheses left (recall that the reduction steps, especially (9), result in cutting off the logically stronger formulas and leaving these logically weaker – thus with great probability, a non-significant hypothesis is simply the abductive goal or a formula equivalent to it; see problems P7 and P37 as examples). What is more, one can see that the only criterion which is in no great conflict with Consistency is Significance – apart from the fact that the number of non-significant hypotheses is low, this is probably in part due to the fact that both of

¹⁰Just to state the obvious, were the set of premises inconsistent, there would be no abductive problem, because with CPC in the background from the inconsistent set of premises it is possible to derive any formula.

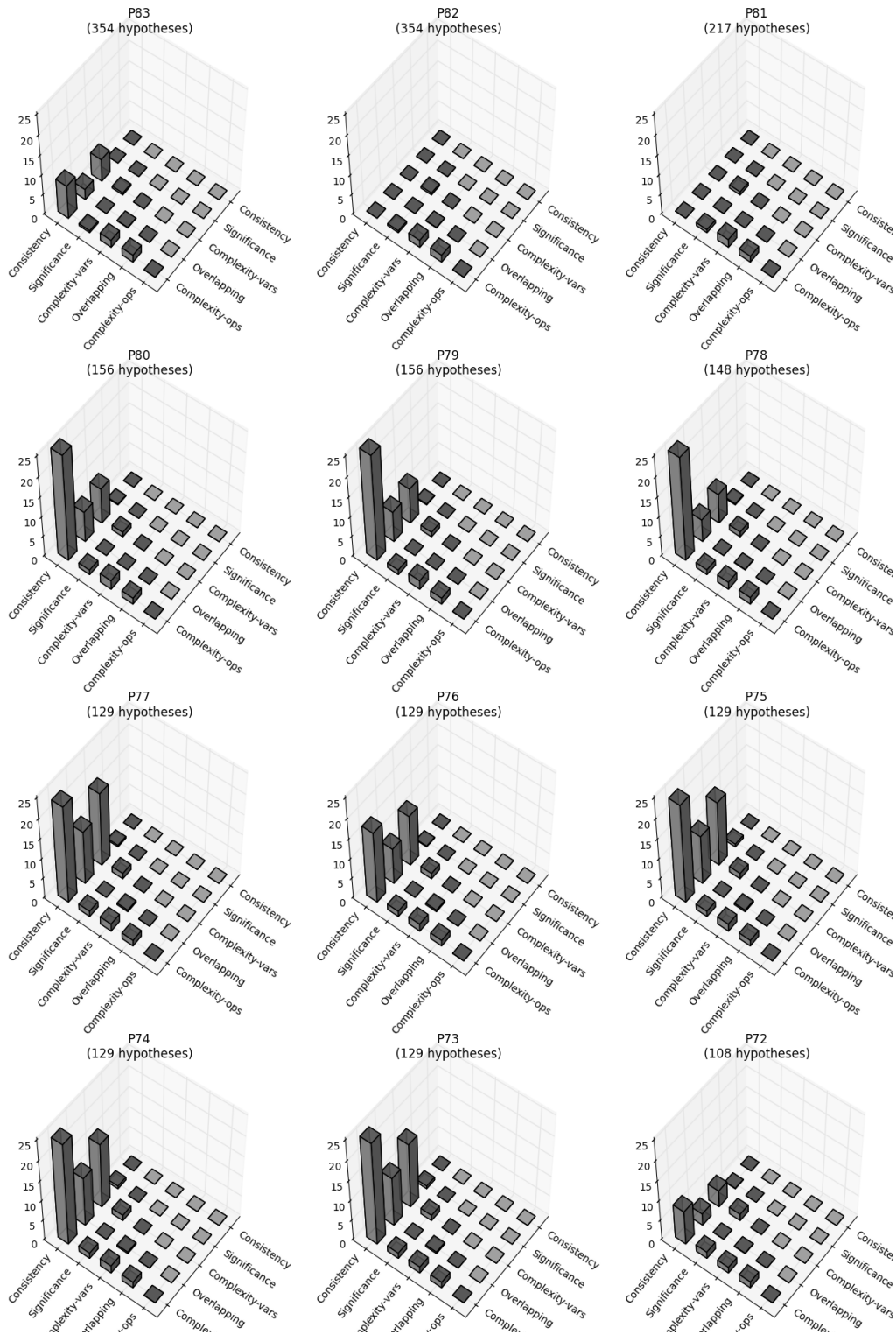


Figure 4: Discordance of criteria (vertical axis in %) for problems with the highest number of unique hypotheses.

these criteria are binary; Overlapping and the two Complexities are multi-valued, which provides more occasions for conflict, thus increasing discordance.

The discordance between Significance and Overlapping is almost non-existent because of the close relation between these two criteria. An abductive hypothesis cannot be logically false and an abductive goal cannot be a tautology, thus a non-overlapping hypothesis (that is, sharing no variables with the abductive goal) cannot entail the abductive goal, hence it is significant. On the other hand, for similar reasons, a non-significant hypothesis must contain the variables of the abductive goal. Overlapping, Complexity and Operational Complexity are all minimized and related to the number of literals occurring in the hypotheses, hence the low level of disagreement between these criteria.

5 Conclusions

The research presented in this article concerned automation of an abductive procedure. As a testbed, a basic logical system – the Classical Propositional Calculus (CPC) – has been chosen to specify the main components of the abductive problems and the abductive procedure. The choice of this simple logic allows one to avoid ambiguities pertinent to more complex logical systems and to focus on the problem of abductive reasoning. On the other hand, employing the proposed approach with the CPC opens a way to use it in more sophisticated systems such as first-order logic, since the considerations reported in this work are of a more general character.

In order to define the procedure of generating abducibles we employed a proof method called the Synthetic Tableau Method (STM). For each one of the analyzed problems, information present on suitable branches of synthetic tableaux (that is, the so-called entangled literals of synthetic failures) was used to form abductive hypotheses. Because of a potentially extensive number of the candidate abducibles, we developed a system enabling reduction, simplification and selection based on a multi-criteria evaluation of the abductive hypotheses.

The results of our research demonstrate that filtering out logically equivalent and/or logically too strong abducibles using various reduction and simplification techniques decreases sizes of sets of hypotheses by many orders of magnitude. Furthermore, the multi-criteria evaluation enabled reduction of the number of abducibles to the ones that are more appealing than others according to the criteria considered. As we have shown, the employment of the dominance relation provided a flexible and an assumption-free method to select interesting abducibles, conforming to several criteria at the same time, and depending on other available hypotheses in the set. The combination of reduction and simplification with the dominance-based evaluation proved to be highly efficient in finding simple abductive hypotheses which solve problems in agreement with human intuitions and deductive experience.

The multi-criteria dominance relation approach allowed measuring interactions between evaluation criteria of abductive hypotheses. As it has been demonstrated, interactions between criteria may reveal the underlying structure of a problem. The combination of refined analyses of such an interplay in more sophisticated logical systems, with massive problem sampling and the use of advanced data mining tools, is a promising direction of research. This approach may not only contribute to understanding of a complex structure of abductive problems, but may also be utilized in automated generation of problem-solving heuristics. This interesting issue will be further investigated.

The aim of our research was to develop an automated system dedicated to solve abductive problems in an exemplary logical system. Following encouraging results of this work, we would like to move on to more advanced logics – such as the classical first-order logic or a modal logic. Due to the fact that a combination of logically modeled abduction and artificial intelligence techniques has already been shown to be possible [2, 30], more advanced decision support techniques such as data mining, knowledge discovery and optimization algorithms may be introduced in the system. The process of abduction is probably one of the most complex types of reasoning to be grasped by means of automated logical tools. At the same time, it is challenging to model and effectively simulate this process by logical tools combined with contemporary artificial intelligence methods (cf. [33]). The potential profits of such research are high, as the essence of an abductive process is to make a creative use of available knowledge.

References

- [1] Atocha Aliseda. *Abductive Reasoning. Logical Investigations into Discovery and Explanation*. Springer, Dordrecht, 2006.
- [2] Maurizio Atzori, Paolo Mancarella, and Franco Turini. Abduction in classification tasks. In *AI IA 2003: Advances in Artificial Intelligence*, pages 213–224. Springer, 2003.
- [3] S. R. Buss, editor. *Handbook of Proof Theory*. Elsevier, Amsterdam, 1998.
- [4] Tom Bylander, Dean Allemang, Michael C. Tanner, and John R. Josephson. The computational complexity of abduction. *Artificial intelligence*, 49(1):25–60, 1991.
- [5] C. A. C. Coello, G. B. Lamont, and D. A. van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. Springer, 2007.
- [6] Marcello d’Agostino. Are tableaux an improvement of truth-tables? *Journal of Logic, Language and Computation*, 1:235–252, 1992.
- [7] M. Doumpos and E. Grigoroudis. *Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications*. Wiley, 2013.
- [8] Bradford Dunham and Hao Wang. Towards feasible solutions of the tautology problem. *Annals of Mathematical Logic*, 10:117–154, 1976.
- [9] M. Ehrgott. *Multicriteria Optimization*. Springer, 2006.
- [10] Thomas Eiter and Georg Gottlob. The complexity of logic-based abduction. *Journal of the ACM (JACM)*, 42(1):3–42, 1995.
- [11] Dov M. Gabbay and John Woods. *The Reach of Abduction. Insight and Trial*. Elsevier, 2005.
- [12] F. Glover and G. A. Kochenberger. *Handbook of Metaheuristics*. International series in operations research & management science. Kluwer Academic Publishers, 2003.
- [13] Maciej Hapke and Maciej Komosinski. Evolutionary design of interpretable fuzzy controllers. *Foundations of Computing and Decision Sciences*, 33(4):351–367, 2008.
- [14] Jaakko Hintikka. What is abduction? The fundamental problem of contemporary epistemology. In *Inquiry as Inquiry: A Logic of Scientific Discovery*, pages 91–113. Springer, 1999.
- [15] H.H. Hoos and T. Stützle. *Stochastic local search: Foundations and applications*. Morgan Kaufmann, 2005.
- [16] Katsumi Inoue. Automated abduction. In *Computational Logic: Logic Programming and Beyond*, pages 311–341. Springer, 2002.
- [17] Tarun Kumar Jain, Dharmender Singh Kushwaha, and Arun Kumar Misra. Optimization of the Quine-McCluskey method for the minimization of the boolean expressions. In *Fourth International Conference on Autonomic and Autonomous Systems*, pages 165–168. IEEE, 2008.
- [18] Jacek Jelonek and Maciej Komosinski. Biologically-inspired visual-motor coordination model in a navigation problem. In Bogdan Gabrys, Robert Howlett, and Lakhmi Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 4253 of *Lecture Notes in Computer Science*, pages 341–348. Springer, Berlin/Heidelberg, 2006.
- [19] Maurice Karnaugh. The map method for synthesis of combinational logic circuits. *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, 72(5):593–599, 1953.

- [20] Maciej Komosinski and Adam Kups. Logical abduction experiment definition: script sources and sample usage, 2014. <http://framsticks.com/trac/framsticks/browser/experiments/frams/logic>.
- [21] Maciej Komosinski, Adam Kups, and Mariusz Urbański. Multi-criteria evaluation of abductive hypotheses: towards efficient optimization in proof theory. In *Proceedings of the 18th International Conference on Soft Computing*, pages 320–325, Brno, Czech Republic, 2012.
- [22] Maciej Komosinski and Adam Rotaru-Varga. From directed to open-ended evolution in a complex simulation model. In Mark A. Bedau, John S. McCaskill, Norman H. Packard, and Steen Rasmussen, editors, *Artificial Life VII*, pages 293–299. MIT Press, 2000.
- [23] Maciej Komosinski and Szymon Ulatowski. Framsticks web site, 2014. <http://www.framsticks.com>.
- [24] Lorenzo Magnani. *Abductive Cognition. The Epistemological and Eco-Cognitive Dimensions of Hypothetical Reasoning*. Springer, 2009.
- [25] Marta Cialdea Mayer and Fiora Pirri. First order abduction via tableau and sequent calculi. *Bulletin of the IGPL*, 1:99–117, 1993.
- [26] Marta Cialdea Mayer and Fiora Pirri. Propositional abduction in modal logic. *Logic Journal of IGPL*, 3(6):907–919, 1995.
- [27] E. J. McCluskey, Jr. Minimization of boolean functions. *Bell System Technical Journal*, pages 1417–1444, 1956.
- [28] Joke Meheus and Diderik Batens. A formal logic of abductive reasoning. *Logic Journal of the IGPL*, 14:221–236, 2006.
- [29] Joke Meheus, Liza Verhoeven, Maarten Van Dyck, and Dagmar Provijn. Ampliative adaptive logics and the foundation of logic-based approaches to abduction. In *Logical and Computational Aspects of Model-Based Reasoning*, pages 39–71. Kluwer, 2002.
- [30] Tim Menzies. Applications of abduction: knowledge-level modelling. *International Journal of Human-Computer Studies*, 45(3):305–335, 1996.
- [31] Peter Merz and Bernd Freisleben. *Fitness Landscapes and Memetic Algorithm Design*, pages 245–260. McGraw-Hill, 1999.
- [32] Charles Sanders Peirce. *Collected Works*. Harvard University Press, Cambridge, MA, 1931 – 1958.
- [33] Oliver Ray. Automated abduction in scientific discovery. In *Model-Based Reasoning in Science, Technology, and Medicine*, pages 103–116. Springer, 2007.
- [34] E. G. Talbi. *Metaheuristics: From Design to Implementation*. Wiley Series on Parallel and Distributed Computing. Wiley, 2009.
- [35] Paul Thagard. Abductive inference: From philosophical analysis to neural mechanisms. In A. Feeney and E. Heit, editors, *Inductive reasoning: Cognitive, mathematical, and neuroscientific approaches*, pages 226–247. Cambridge University Press, Cambridge, 2007.
- [36] Mariusz Urbański. Remarks on synthetic tableaux for classical propositional calculus. *Bulletin of the Section of Logic*, 30(4):194–204, 2001.
- [37] Mariusz Urbański. Synthetic tableaux for Łukasiewicz’s calculus Ł3. *Logique et Analyse*, 177-178:155–173, 2002.

- [38] Mariusz Urbański. How to synthesize a paraconsistent negation. The case of CLuN. *Logique et Analyse*, 185-188:319–333, 2004.
- [39] Mariusz Urbański. Tableaux, abduction and truthlikeness. Technical report, Instytut Psychologii UAM, Poznań, 2005.
- [40] Leonardo Vanneschi, Marco Tomassini, Philippe Collard, and Manuel Clergue. Fitness distance correlation in structural mutation genetic programming. In *Proceedings of the 6th European conference on Genetic programming*, EuroGP'03, pages 455–464, Berlin, Heidelberg, 2003. Springer-Verlag.
- [41] Edward W. Veitch. A chart method for simplifying truth functions. In *Proceedings of the 1952 ACM national meeting (Pittsburgh)*, pages 127–133. ACM, 1952.