Artificial Intelligence 246 (2017) 181–219

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Lakatos-style collaborative mathematics through dialectical, structured and abstract argumentation

Alison Pease [a,*], John Lawrence [a], Katarzyna Budzynska [b,a], Joseph Corneli [c,d], Chris Reed [a]

[a] Centre for Argument Technology, University of Dundee, UK
[b] Institute of Philosophy and Sociology, Polish Academy of Sciences, Poland
[c] Department of Computing, Goldsmiths, University of London, UK
[d] University of Edinburgh, UK

ARTICLE INFO

Article history:
Received 9 October 2015
Received in revised form 17 February 2017
Accepted 20 February 2017
Available online 1 March 2017

Keywords:
Automated theorem proving
Automated reasoning
Abstract argumentation
Argumentation
Collaborative intelligence
Dialogue games
Lakatos
Mathematical argument
Structured argumentation
Social creativity
Philosophy of mathematical practice

ABSTRACT

The simulation of mathematical reasoning has been a driving force throughout the history of Artificial Intelligence research. However, despite significant successes in computer mathematics, computers are not widely used by mathematicians apart from their quotidian applications. An oft-cited reason for this is that current computational systems cannot do mathematics in the way that humans do. We draw on two areas in which Automated Theorem Proving (ATP) is currently unlike human mathematics: firstly in a focus on soundness, rather than understandability of proof, and secondly in social aspects. Employing techniques and tools from argumentation to build a framework for mixed-initiative collaboration, we develop three complementary arcs. In the first arc – our theoretical model – we interpret the informal logic of mathematical discovery proposed by Lakatos, a philosopher of mathematics, through the lens of dialogue game theory and in particular as a dialogue game ranging over structures of argumentation. In our second arc – our abstraction level – we develop structured arguments, from which we induce abstract argumentation systems and compute the argumentation semantics to provide labelings of the acceptability status of each argument. The output from this stage corresponds to a final, or currently accepted proof artefact, which can be viewed alongside its historical development. Finally, in the third arc – our computational model – we show how each of these formal steps is available in implementation. In an appendix, we demonstrate our approach with a formal, implemented example of real-world mathematical collaboration. We conclude the paper with reflections on our mixed-initiative collaborative approach.

1. Introduction

The simulation of mathematical reasoning has been a driving force throughout the history of Artificial Intelligence research [58,86,87,98]. However, despite significant successes in 'computer mathematics' (e.g., [18,40,42,45]) computers are not widely used by mathematicians apart from their quotidian applications like running word processing tools, email programs, web servers and web browsers, and (sometimes) computer algebra systems. An oft-cited reason for this is that current computational systems cannot do mathematics in the way that humans do. Despite – or perhaps because of [69] –

* Corresponding author.
E-mail address: a.pease@dundee.ac.uk (A. Pease).

http://dx.doi.org/10.1016/j.artint.2017.02.006

their profound rigour, machine proofs are often thought to be unclear, uninspiring and untrustworthy, as opposed to human proofs which can be deep, elegant and explanatory [21,41]. In order to help to close the gap between machine-constructed proofs and human-constructed ones, we consider two key areas of focus: informal and social aspects of proof discovery in the human context. We propose that theories and tools from the field of argumentation can be used to more closely align AI systems with the human context in these two areas.

1.1. Informal aspects of proof

Evaluation metrics in the Automated Theorem Proving (ATP) community are focused on soundness, and the power of a solver to prove a wide selection of difficult problems with specific resource limits.[1] Qualities of the resulting proof other than soundness are rarely considered. This stands at variance with the practices of the mathematical community, in which a lack of soundness might be forgiven if a proof is interesting or complex. Indeed, an error in a proof may be neither "perturbing," nor "surprising," if it is judged to be the right sort of error (one which is not critical to the integrity of the proof) [20].[2] Instead, one of the main criteria by which a proof is judged in the human context is its understandability. A well-written proof can provide insight as to why a theorem may be true, point to new conjectures, form connections between different fields and suggest solutions to open problems [44,75,106]. Fields medal winners Gowers and Thurston, respectively, have said: "We like our proofs to be explanations rather than just formal guarantees of truth" [41, p. 3], and "reliability does not primarily come from mathematicians formally checking formal arguments; it comes from mathematicians thinking carefully and critically about mathematical ideas" [94, p. 10]. Thurston emphasises that informal conversations between mathematicians can often convey ideas more quickly and comprehensibly than a written proof [94, p. 6]. Hersch has suggested that "The standard style of expounding mathematics purges it of the personal, the controversial, and the tentative, producing a work that acknowledges little trace of humanity, either in the creators or the consumers" [47, p. 131].

Lakatos offered similar insight into proof-understanding [55]. Building on Pólya's distinction between informal, unfinished mathematics-in-the-making and formal, finished mathematics [76], he argued that a theorem and proof which are presented in isolation from their development are "artificial and mystifyingly complicated", analogous to a "conjuring act" [55, p. 142]. In order to make results understandable, they should be presented alongside the "struggle" and "adventure" involved in the story of their development. This insight is echoed by Ernest, who criticises the practice of presenting mathematics learners with the "sanitized outcomes of mathematical enquiry": "The outcome may be elegant texts meant for public consumption, but they also generate learning obstacles through this reformulation and inversion" [67, p. 67]. Bundy points out that this practice also obscures understandability for research mathematicians: "Mathematicians find informal proofs more accessible and understandable [than formal proofs]" [21, p. 2].

In contrast to the concerns about understandability voiced by mathematicians and philosophers of mathematics, understandability is not traditionally a concern for ATP. A handful of exceptions have focused on making an existing machine proof more comprehensible [29,30,36]. MacKenzie [60] has argued that rather than treating machines as oracles and giving them responsibility for verifying the reliability of hardware and software, there needs to be a continued interaction between computer systems and our collective human judgement: "The finished product of formal verification – the 'proof object' – may thus be less important than the process of constructing it." [61, p. 2348]. Constructing or verifying proofs which are written in a classical logical formalism does not align with mainstream mathematical activity, since proofs are typically neither constructed nor presented in this way.

Accordingly, our objective to model mathematical dialogues connects closely with the theory of defeasible argument (reasoning that is rationally compelling but not deductively valid [52]). The structure of classical proof theoretic systems and formal theorisations of defeasible argument differ [99]. Defeasible argument is used during the initial construction of a proof, and as the proof is refined or changed over time to reflect conceptual changes in the underlying theory, or to rectify deductive errors discovered after a proof is commonly accepted – all themes that Lakatos emphasised [55]. In practice, we may have an argument whose conclusion states that *for all* x, $P(x) \rightarrow Q(x)$, whose logical validity rests on a particular interpretation of P and Q. In some cases P or Q might not be clearly defined, and can be subsequently defined in different ways by different people, sometimes rendering the initial argument invalid. Whether a consensus ever occurs and whether we could be sure that the consensus is final, is an open and somewhat contentious question.

We propose that applied argumentation theory can improve the understandability of output from, and input to, ATP systems, and other computer-mediated, -moderated, or -motivated proof systems. Doing this will help to close the cultural gap between human and machine mathematics. One way to go about this is to keep track of informal proof development, presenting the errors, conflicts and deadends involved, alongside a finished or current proof artefact.

1.2. The social dimension of human mathematics

The social dimension is typically neglected in automated reasoning, which usually consists of two approaches: *autonomous theorem proving*, in which a single system proves theorems, or *interactive theorem proving*, in which there is one

[1] "Wide", "difficult" and "resources" are all defined appropriately: see, for instance [92].

[2] Indeed, Aschbacher – one of the main mathematicians involved in the development of the proof of the Classification of Finite Simple Groups (one of the main achievements of twentieth century mathematics, on which many other results depend) – commented that "the probability of an error in the [CFSG] proof is one" [9], related in [20].

system and one user interacting with it. New models of working in a social context have gained traction with the notion of the social machine – a new paradigm identified by Berners-Lee and Fischetti [13] for viewing a combination of people and computers as a single problem-solving entity. Martin and Pease propose a research agenda for mathematics social machines as "a combination of people, computers, and mathematical archives to create and apply mathematics, with the potential to change the way people do mathematics, and to transform the reach, pace, and impact of mathematics research" [63, p. 1]. Epstein outlines a more general vision of collaborative intelligence, in which she revisits original goals in AI in the light of its history, successes and failures, and recommends a new form of synergy between people and computers [34].

These ideas suggest a third approach in automated reasoning – *mixed-initiative proving* – in which proof discovery occurs via interaction of multiple participants (both human and computer) working together towards a common goal. In this paper we build on these ideas and use techniques from argumentation to provide a way of formalising social aspects of mixed-initiative proof via dialogue theory.

1.3. Argumentation

The relationship between the study of mathematical practice and argumentation theory is not well-explored, though it has already borne some fruit. Toulmin applied his argumentation structure to Theaetetus's proof that there are exactly five platonic solids [95]; Aberdein showed that Toulmin's structure can represent more complex mathematical proofs [2,3]; and Alcolea has shown that it can also be used to represent meta-level mathematical argument, such as axiom adoption or rejection [8]. Krabbe [53], Aberdein [4–6] and Aberdein and Pease [72] further demonstrate the application of various theories of argumentation to mathematical argument. Further related issues are explored in a recent edited volume [7].

Although theories of argument have been applied within the philosophy of mathematical practice (as above), mathematics presents a largely novel domain of discourse within the argumentation research community, perhaps due to the misconception that it represents a deductive style of reasoning, more appropriate to formal proof than argumentation.

1.4. Aims and contributions

Our main aim is to develop a new approach in which tools and theories from the argumentation community can be deployed to build a bridge between interactive proof tools and human mathematicians.

Our main contribution is to identify two areas in which ATP is currently unlike human mathematics – informal and social aspects – and to employ techniques and tools from argumentation to build a framework which opens the door to mixed-initiative collaborative reasoning in mathematics. Specifically, we:

1. Propose and demonstrate a way to make proofs more understandable by drawing on philosophical, sociological and educational literature on mathematics which highlights the importance of presenting the development of a proof attempt alongside a final, or currently accepted, proof artefact. We develop a framework in which this aspect is possible. A grounded extension of a dialogue can be produced, representing a currently accepted, collaboratively constructed, proof or theory. Since the record of the dialogue can be presented alongside this proof, the framework delivers the history of a proof attempt as well as the proof artefact.
2. Propose and demonstrate a way to make collaboration more social by opening the door to a mixed initiative collaborative mathematics. Social aspects in human mathematics have been shown to be integral to the human context, therefore technologies which are able to support mathematicians in the collective construction of mathematical knowledge, in a variety of ways are essential. In addition to being able to show the current state of discussion (as discussed above), this includes highlighting conflicting commitments or unresolved moves, finding similarities and conflicts across different discussions going on in parallel among otherwise independent groups of arguers, storing past discussions and making them searchable, and so on.
3. Develop three complementary arcs:
   (a) The first arc comprises our *theoretical model*. Starting from the philosophical stance provided by Lakatos [55] in his account of the dialectical, collaborative interaction that constitutes the practice of mathematics, we interpret this through the lens of dialogue game theory [43] and in particular as a dialogue game ranging over structures of argumentation.
   (b) In the second arc we develop an *abstraction level*, in which we start with the argument fragments created via the formal dialogue game and develop structured arguments, from which we induce abstract argumentation systems in the style of [32]. The final stage in this arc is to compute the argumentation semantics, to provide labellings of the acceptability status of each argument. This paper shows how the labelling derived from the abstract argumentation framework corresponds precisely to the theory that has been collaboratively created by the participants in a Lakatosian dialogue. Thus the output from this stage corresponds to a final, or currently accepted proof artefact, which can be viewed alongside its historical development.
   (c) The third arc comprises our *computational model*, in which we show how each of these formal steps is available in implementation. The interpretation of Lakatos as a formal dialogue game can be captured as an implemented specification in the Dialogue Game Description Language (DGDL) [105]. This specification can be executed by a platform, the Dialogue Game Execution Platform (DGEP) [15] which offers a series of web services to clients for

executing a participant's legal moves. Part of the semantics of the dialogue game specification is to define updates on a shared information state [96], in which the language of knowledge representation is AIF, implemented as a series of web services provided by the AIFdb infrastructure [57]. The AIF data created as a side effect of the operational semantics of turn-taking in the dialogue game is interpreted by The Online Argument Structures Tool (TOAST) [90] as an ASPIC+ system [66] and passed to DungOMatic [88] to calculate the grounded extension. At each step in the game, this calculation returns the current state of the co-created mathematical theory.

4. Demonstrate our approach and the interaction of our three arcs via an implemented example of human mathematical collaboration. This shows how off-the-shelf technologies can produce a pipeline system, running from natural language dialogue about the construction of a mathematical proof, to philosophical theory, to the formal expression of natural language reasoning in dialogue games, and from there to abstract argumentation and argumentation semantics, and finally, coming full circle, to show that implementations of these systems can then provide value back to the mathematical community from which the philosophical theory was derived.
5. Demonstrate the applicability of argumentation techniques to mathematical reasoning.

We further show how the model can be retrospectively applied to examples of extant mathematical discussion in Appendix A. In so doing, it is not only possible to demonstrate the depth of Lakatos's original insight, but also to show that the formal characterisation here remains both honest to the original and of practical utility to mathematicians. By making this connection back to the community of mathematical practice, we show that the door is opened to mixed-initiative, collaborative mathematics [89,35].

Prior work on argumentation in artificial intelligence is surveyed in [12] and [11]. Several 'pipeline' systems for argument analysis have previously been developed [107,88], "breaking down argumentation tools into small components, deploying these components as web services, then constructing UNIX-style pipelines to link them together as one large system" [88, p. 1]. The specific 'pipeline' detailed in the later sections of this paper is outlined in Fig. 1c. We emphasise logic rather than linguistics, and thus do not solve the problem of the "knowledge acquisition bottleneck" highlighted by Wyner, van Engers, and Hunter [107, p. 21]. Section 8 points to some related recent work on the linguistics of mathematics that could be useful in widening this bottleneck. Indeed, there are at least two narrow places in the bottleneck – and the corresponding research challenges are roughly analogous to the steps "first from the established facts to intermediate predicates, and then from these intermediate predicates to legal consequences" in legal case-based reasoning [81, p. 17]. The present work is unique in that it is formal, implemented, and descriptive of real-world mathematical collaboration.

The remainder of the paper is structured as follows: in Sections 2–3 we outline our theoretical model, in which we introduce theoretical foundations and develop a formal dialogue system from Lakatos's model of mathematical discourse. In Sections 4–5 we present our abstract level, showing how we progress from a formal dialogue system to Argument Interchange Format structures and then to abstract argumentation frameworks. In Sections 6–7 we present our computational model and example of collective proof as argumentation. Finally, we present our conclusions in Section 8. A diagrammatic representation of the paper can be found in Fig. 1.

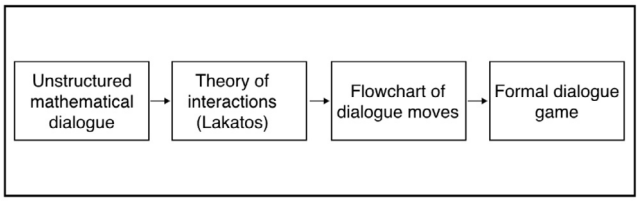
2. Theoretical foundations

In this section we lay down the theoretical foundations necessary for understanding the rest of the paper. Section 2.1 contains a high-level description of Lakatos's patterns of dialogue: this, alongside material in Section 2.3 on specifying dialectical interactions as a set of rules governing the interaction, will be needed to understand our development of a formal dialogue game in Section 3. Section 2.2 shows the application of Lakatos's patterns to real-world natural language discourse in mathematics, demonstrating their applicability to at least some human mathematical reasoning (more detailed applicability is demonstrated in Appendix A). Section 2.4 contains background on the Argument Interchange Format, and we review structured argumentation in Section 2.5: we draw on both of these in Section 4. Finally, in Section 2.6 we describe abstract argumentation, which we use in Section 5. All sections will be relevant for understanding our implementation work and the execution of the implemented system described in Sections 6 and 7.

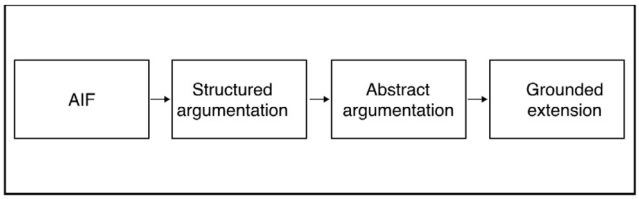
2.1. Lakatos's patterns of dialogue

Lakatos has contributed to the field of AI both methodologically in his philosophy of science [54] and via his ideas on the growth of informal mathematics [55]. Against the traditional view that mathematical progress comes down lucky guess work or simple intuition, Lakatos proposed that there are logical mechanisms – albeit only informally specified in his work – which underly the mathematical thought process. He challenged Popper's view [77] that philosophers can form theories about how to evaluate conjectures, but not theories about how to generate them. He did this in two ways: arguing that *(i)* there is a logic of discovery, *i.e.*, the process of generating conjectures and proof ideas is subject to rational laws; and, *(ii)* a sharp distinction between discovery and justification is misleading as each affects the other.

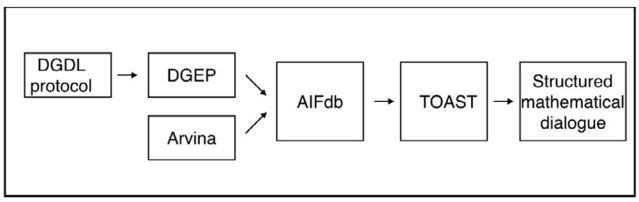
We know of two systematic implementations of Lakatos's philosophy of mathematics. Pease [71] constructed a multi-agent system in which agents formed and communicated theories in both mathematical and non-mathematical domains, and responded by following Lakatos's patterns of dialogue. Hayes-Roth [46] developed heuristics for repairing flawed plans, modelled on [55] and developed in the context of a card game.



(a) 'Theoretical model/analysis' (Sections 2-3)



(b) 'Abstraction level' (Sections 4-5)



(c) 'Computational models/analyses' (Sections 6-7)

Fig. 1. The three main developmental arcs discussed in the paper.

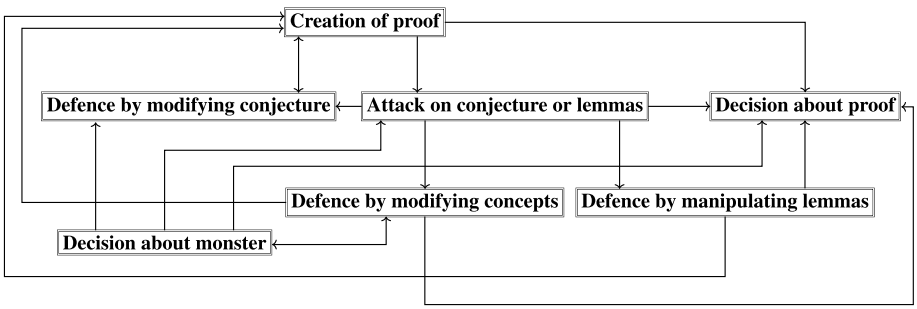


Fig. 2. Lakatos's informal logic of mathematical discovery [55], represented as a flow chart.

At a higher level, Sloman [87] highlights the relevance of Lakatos's notions to AI, via his analysis of the capabilities that would be necessary for an intelligent robot (or young child, or other human) to think mathematically. Sloman argues that, since such thinking is not infallible in humans, the capacity to discover and repair flaws in arguments and conclusions is essential. Lakatos's fallibilist picture of mathematics provides a detailed account of how this can be done, by outlining various methods by which discovery of mathematical claims and their justification in the form of an argument can occur. These methods suggest ways in which concepts, conjectures and proofs gradually evolve via interaction between mathematicians. In Lakatos's account – which he presented as a classroom discussion between (very advanced) students – proofs, conjectures and concepts are fluid and open to negotiation.

Lakatos demonstrated his argument by presenting case studies in dialogue form of the development of Euler's conjecture that for any polyhedron, the number of vertices (V) minus the number of edges (E) plus the number of faces (F) is equal to two, and Cauchy's proof of the conjecture that the limit of any convergent series of continuous functions is itself continuous. The dialogue highlights various moves that can be taken at different times, and the consequences of these moves on the resulting theory. We outline these dialogue moves below and represent the flow in Fig. 2 This figure serves as a visual key to the formalisation developed in Section 3. The example in Section 7 is a portion of Lakatos's fictional debate of Euler's conjecture.

The first step in a Lakatosian dialogue is for someone to **propose a conjecture**. This is optionally followed by a proof, consisting of a list of lemmas and a statement that the conjecture is now proved. To emphasise: for Lakatos, the basic structure of a mathematical proof is nothing other than a list of lemmas together with a conclusion that they support.

After this, participants might **accept the conjecture**, which will terminate the dialogue. Alternatively, they might perform **strategic withdrawal**, which weakens the conjecture itself, but (ideally) strengthens confidence in it. If this move is used, then the subsequent options for dialogue moves are those which are available after the proposal of the initial conjecture. Strategic withdrawal consists of using positive examples of the conjecture and generalising from these to a class of object, and then limiting the domain of the conjecture to this class. For instance, the students generalise from regular polyhedra to *convex polyhedra*, and then modify Euler's conjecture to 'for any convex polyhedron, $V - E + F = 2$'. A third alternative at this stage is for a participant (opponent) to challenge the conjecture by **raising a counterexample**. When a counterexample has been raised, participants (specifically, proponents) have five moves available to them.

Firstly, they can **surrender** the conjecture, which will terminate the dialogue.

Secondly, they can deal with exceptions via **piecemeal exclusion** and thereby exclude a whole class of counterexamples. Piecemeal exclusion amounts to generalising from a counterexample to a class of counterexamples which have certain properties and then excluding the entire class. Again, in terms of dialogue moves, this is a form of **propose a conjecture** (which will be a weakened version of the original conjecture), and subsequent dialogue moves will follow the same pattern as for this move. An example of piecemeal exclusion is that the students generalise from the hollow cube to *polyhedra with cavities*, and then modify Euler's conjecture to 'for any polyhedron without cavities, $V - E + F = 2$'.

Thirdly, participants can perform **monster-barring**. This is a way of excluding an unwanted counterexample, and consists of the argument that the proposed 'counterexample' is not valid, as it is not within the claimed concept definition (this may then be expanded). Thus, it does not conflict with the conjecture, because it is *not* a counterexample. For instance, one of the students suggests that the hollow cube (a cube with a cube-shaped hole in it) is a counterexample to Euler's conjecture, since $V - E + F = 16 - 24 + 12 = 4$. Another student uses monster-barring to argue that the hollow cube does not threaten the conjecture as it is not in fact a polyhedron. The concept polyhedron then becomes the focus of the discussion, with the definition possibly being formulated explicitly for the first time. Thus, subsequent moves here must either propose an alternative definition, or express a preference to participants as to which definition should be adopted. Using this method, the original conjecture is unchanged, but the meaning of the terms in it may change.

Fourthly, participants can perform **monster-adjusting**, which is similar to monster-barring. Here, one *reinterprets* an object in such a way that it is no longer a counterexample, not by (re)defining the domain of the conjecture (so in this case the object is still seen as belonging to this domain), but by (re)defining subconcepts in the conjecture. Subsequent moves here are analogous to those following monster-barring, for appropriate concept definitions. The example in [55] concerns the star polyhedron. This entity is raised as a counterexample since, it is claimed, it has 12 faces, 12 vertices and 30 edges (where a single face is seen as a star polygon), and thus $V - E + F$ is -6. This is contested, and it is argued that it has 60 faces, 32 vertices and 90 edges (where a single face is seen as a triangle), and thus $V - E + F$ is 2. The argument then turns to the definition of 'face': again, using this method, the original conjecture is unchanged, but the meaning of the terms in it may change.

Finally, a fifth dialogue move when faced with a counterexample is to perform **lemma incorporation**. This works by considering the counterexample and determining whether it is global (a counterexample to the main conjecture) and/or local counterexamples (a counterexample to one of the lemmas). If it is both global and local, *i.e.*, there is a problem both with the argument and the conclusion, then the move consists in modifying the conjecture by incorporating the problematic proof step as a condition. If it is local but not global, *i.e.*, the conclusion may still be correct but the reasons for believing it are flawed, then the move consists in modifying the problematic proof step but leaving the conjecture unchanged. If it is global but not local, *i.e.*, there is a problem with the conclusion but no obvious flaw in the reasoning which led to the conclusion, then the move consists in looking for a hidden assumption in the proof step, then modifying the proof and the conjecture by making the assumption an explicit condition. For example, in the portion of the Euler conjecture discussion treated in Section 7, the Teacher remarks with respect to a local counterexample: "I no longer contend that the removal of any triangle follows one of the two patterns mentioned, but merely that at each stage of the removing operation the removal of any boundary triangle follows one of these patterns…I can easily improve the proof, by replacing the false lemma by a slightly modified one, which your counterexample will not refute." The discussion can then continue in the same fashion as when the initial problem and proof were proposed.

2.2. Analysing natural language dialogues

Dialogue has been a topic of interest for computer scientists for over three decades [50,48]. Dialogue is, fittingly, approached through NLP and linguistic analysis [37], but it also falls within the scope of pragmatics [22], which "arises as soon as we move beyond the linguistic analysis of an utterance and ask what the speaker meant by it" [91]. Searle's theory of speech acts [85] is a classic in this genre. Pragmatic analysis is relevant to both human–computer and multi-agent collaboration [51,74]. *Dialogue game* protocols have previously been applied, for example, to multi-agent planning problems [10,17]. Dialogue games are "more expressive than auction and game-theoretic mechanisms, typically allowing participants to question and contest assertions, to advance supporting arguments and counter-arguments, and to retract prior assertions" [65]. Given their formality, it is not surprising that, at least by default, "there is no direct connection between formal di-

alogue games or a theory of sentence meaning and natural language use" [104, p. 81]. Thus, alongside application areas like collaborative planning and proof construction, research on natural language dialogues is associated with a distinct collection of fundamental communication issues, such as modelling the ways in which mutual belief is established between discussants [97]. For example, the *contribution model* of Clark and Schaefer [26] is centred on building common understanding through iterated phases of *presentation* and *acceptance*. However, Lakatos's informal logic outlined in Fig. 2 assumes that discussants are able to understand each other satisfactorily at the linguistic level, by and large, and focuses instead on building a shared mathematical theory. The Lakatos Game developed in Section 3 expands the schematic sketched in Fig. 2 in detail. In Appendix A, we use the full range of this model to mark up a real-world example of a collaboratively-constructed proof. This shows the applicability of Lakatos's theory via our interpretation of his work as a dialogue game, and, more broadly, the relevance of our overall approach, which we outline in further detail below.

2.3. Formal dialogue systems

The pipeline of collaborative mathematics starts with the expressing the Lakatos model as a formal dialogue system. The approach of specifying dialectical interaction as a game or a set of rules governing the interaction has started with the work of Lorenzen [59] and Hamblin [43]. Lorenzen aimed to translate a system of logic (such as intuitionistic logic or classical logic) into a system of dialectical rules, the dialogical logic, in which the proponent and the opponent aim to collectively prove that a formula is the tautology of this logic (intuitively, that it is true in this logic). Hamblin, on the other hand, tried to show that at least some fallacies (flaw argumentation patterns) have a dialectical nature even though they have inferential structure. He designed a formal dialogue system, called formal dialectics, and demonstrated that it is possible to formulate rules which will prohibit the players to commit the fallacy of circular reasoning of the form "p because p". This work led to a variety of similar systems which improved the original ones, responded to other philosophical problems, or model the communication amongst agents in multi-agents systems (see e.g. [102,64,68,105]). In this paper we will treat these systems as a template for the development of formal dialogue systems.

In the literature, many different types of dialogues were identified (see e.g. [103] for the first and widely used attempt of classification of dialogues). Lakatos describes the interactions of mathematicians, when aiming to prove the conjecture, in the way that is the most closely related to the concept of persuasion dialogue, and in particular – its "conflict resolution" subtype [19]. Persuasion dialogue is triggered by a difference of opinion between participants, each of them aims to persuade each other and the main goal of the conversation is to achieve the resolution of the conflict.

An excellent survey of systems for persuasion is given in [79]. A dialogue systems has a dialogue purpose, a set A of participants and a set R of roles which participants can adopt during a game. Contents of utterances used by the players in the dialogue are expressed in a topic language L_t. At the beginning of a dialogue every player s has assigned a (possible empty) set of commitments $C_s \subseteq L_t$ which changes during a dialogue. Every dialogue system includes a logic L consisting of a topic language L_t and a set R of inference rules over L_t. The dialogue system consists of several sets of rules, amongst which the most typically used there are: **(1)** locution rules which describe what type of utterances players can execute during a dialogue; **(2)** structural rules or protocol which determine the interaction between locutions (i.e., it specifies which locution can be performed as a reply to another locution)[3]; **(3)** commitment or effect rules which specify for each utterance φ the effects which this locution makes on a set of commitments of the participant i (a commitment of i is a sentence that i publicly declared as his belief)[4]; **(4)** termination rules which determine the cases where no move is legal, i.e. they should specify the conditions under which the protocol returns the empty set; and **(5)** outcome rules which define the outcome of a dialogue, i.e. provide a criterion to decide which player wins and which player loses the dialogue.

The typical set of locutions and the reply structure identified in [79] is presented in Table 1. There are six legal moves that formal dialogue systems for persuasion often permit the players to perform: claim φ; why φ; concede φ; retract φ; φ since S; and question φ. The structural rules typically allows the players to interact in the following way: for example, after the agent claims a proposition φ his respondent can challenge this proposition, claim it's negation or agree with the speaker; when the agent challenges φ his respondent can justify this statement with a set of propositions S or withdraw the statement; and so on.

This standard of formal representation of dialogical interaction will be used in Section 3 for expressing Lakatos model as a set of rules for mathematicians to follow, if they aim to collectively prove a conjecture.

2.4. Argument Interchange Format

The Argument Interchange Format [25] is an attempt to bring together a wide variety of argumentation technologies so that they can work together. [82] reviews some of the more recent applications of the AIF. Descriptions of the AIF are given in a number of places, as are reifications in languages such as RDF and OWL [25,83,82]. AIF uses a graph-theoretic basis for defining an "upper" ontology of the main components (or nodes) of arguments. Nodes are distinguished into those that capture information (loosely, these correspond to propositions), and those that capture relations between items of information,

[3] Formally, let M be a set of moves. The set of finite dialogues $M < \infty$ is the set of all finite sequences $m_1, ..., m_i$ from M. A protocol, specifying the legal moves at each stage of a dialogue, is a function $P : Pow(L_t) \times D \longleftrightarrow Pow(L_c)$ where $D \subseteq M < \infty$. The elements of D are called the legal finite dialogues.

[4] The function C_i for a sequence of moves assigns a set of commitments.

Table 1
A set of protocol rules typical for formal dialogue systems for persuasion (according to [79]).

Locutions	Replies
claim φ	why φ, claim $\neg\varphi$, concede φ
why φ	φ since S (alternatively: claim S), retract φ
concede φ	
retract φ	
φ since S	why ψ ($\psi \in S$), concede ψ ($\psi \in S$)
question φ	claim φ, claim $\neg\varphi$, retract φ

including relations of inference (which correspond to the application of inference rules to particular sets of propositions), relations of conflict (which represent forms of incompatibility between propositions) and relations of preference (which represent value orderings applied to particular sets of propositions). The instantiated nature of these relations is emphasised in the nomenclature, so whilst information is captured in Information (I-) nodes, relations between them are captured as Rule Application (RA-) nodes, Conflict Application (CA-) nodes and Preference Application (PA-) nodes. The general forms or patterns that these applications instantiate are given in a second part of the AIF ontology, the Forms ontology. The approach follows in the philosophical tradition of Walton [100], [101] of stigmatising stereotypical patterns of reasoning – and then extending the tradition into conflict and preference. It is this schematic underpinning which gives the collective name for RA-, CA- and PA-nodes: Scheme (S-) nodes. The AIF upper ontology is designed to allow specialisation and extension to particular domains and projects, in an attempt to balance the needs of interchange against the needs of idiosyncratic development.

The AIF standard will be used in Section 4 for showing how the execution of rules of Lakatos dialogue game is creating argument maps which represent a collective proof as a directed graph.

2.5. Structured argumentation

Structured argumentation aims to "give a general structured account of argumentation that is intermediate in its level of abstraction between concrete logics and the fully abstract level, providing guidance on the structure of arguments, the nature of attacks, and the use of preferences, while at the same time accommodating a broad range of instantiating logics and allowing for the study of conditions under which the various desirable properties are satisfied by these instantiations" [66, p. 31]. One of the foremost and most flexible accounts of structured argumentation is available in *ASPIC*$^+$ [80,66] which is not only flexible about the logic used to instantiate arguments within it, but also provides a straightforward mechanism for inducing abstract argumentation frameworks, described in the next section.

We simplify the definition in [80] in three ways:

(i) skipping preference ordering over rules, i.e. constraining $\preceq = \preceq = \emptyset$;
(ii) ignoring strict rules (of which there are none permitted in Lakatos' theory), i.e. constraining $\mathcal{R}_s = \emptyset$ and thereby $\mathcal{R} = \mathcal{R}_d$; and
(iii) ignoring non-*ordinary* types of premises, i.e. constraining $\mathcal{K}_a = \mathcal{K}_i = \mathcal{K}_n = \emptyset$ and thereby $\mathcal{K} = \mathcal{K}_p$.

Whilst these are important features of *ASPIC*$^+$ in general, they are not exploited here, and so dropped for clarity. An *ASPIC*$^+$ *argumentation system* is thus a triple $AS = (\mathcal{L}, ^-, \mathcal{R})$, where $\mathcal{L}$ is a logical language, $^-$ is a contrariness function from $\mathcal{L}$ to $2^{\mathcal{L}}$, and $\mathcal{R}$ is a set of rules of the form $\phi_1, \ldots, \phi_n \Rightarrow \phi$. An *ASPIC*$^+$ *argumentation theory* is then a pair $AT = (AS, \mathcal{K})$ where $\mathcal{K}$ is a knowledge base in AS.

An *ASPIC*$^+$ argumentation theory yields an *argument*, A, in two cases:

(i) A is ϕ iff $\phi \in \mathcal{K}$, and in this case, $Prem(A) = \{\phi\}$, $Conc(A) = \phi$, $Sub(A) = \{\phi\}$, $DefRules(A) = \emptyset$ and $TopRule(A) = undefined$;
(ii) A is $A_1, \ldots, A_n \Rightarrow \psi$ iff $A_1, \ldots, A_n$ are arguments and there exists a rule

$$Conc(A_1), \ldots, Conc(A_n) \Rightarrow \psi$$

in $\mathcal{R}$, and in this case,

$$Prem(A) = Prem(A_1) \cup \ldots \cup Prem(A_n),\ Conc(A) = \psi$$

$$Sub(A) = Sub(A_1) \cup \ldots \cup Sub(A_n) \cup \{A\},$$

$$DefRules(A) = DefRules(A_1) \cup \ldots \cup DefRules(A_n) \cup \{Conc(A_1), \ldots, Conc(A_n) \Rightarrow \psi\}$$

and $TopRule(A) = Conc(A_1), \ldots, Conc(A_n) \Rightarrow \psi$.

Finally, an *ASPIC*$^+$ argumentation theory yields an *attack* (A, B) in three cases:

(i) argument A *undercuts* argument B (on B') iff $Conc(A) \in \overline{B'}$ for some $B' \in Sub(B)$ of the form $B''_1, \ldots B''_n \Rightarrow \psi$;
(ii) argument A *rebuts* argument B (on B') iff $Conc(A) \in \overline{\psi}$ for some $B' \in Sub(B)$ of the form $B''_1, \ldots B''_n \Rightarrow \psi$;
(iii) argument A *undermines* argument B (on ψ) iff $Conc(A) \in \overline{\psi}$ for some $\psi \in Prem(B)$.

An *ASPIC*$^+$ argumentation theory thus yields a set of arguments and a set of attacks, which can be considered as an abstract argumentation framework.

2.6. Abstract argumentation

Abstract argumentation provides a mechanism for reasoning over directed graphs in which vertices are propositional labels corresponding to arguments, and edges between them are relationships of conflict or attack. Dung's original paper [32] rests upon a notion of acceptability of arguments which in turn is used to define (amongst others) a set of arguments which might be sceptically believed: the grounded extension. We adapt here the definition from [66] which is particularly concise.

An *abstract argumentation framework*, AF is a pair (AR, Att) where AR is a set of arguments and $Att \subseteq AR \times AR$ is a binary relation of attack. A *semantics* for AFs returns sets of arguments called *extensions*, which are internally coherent and defend themselves against attack. For an AF (AR, Att), for any $X \in AR$, X is *acceptable* with respect to some $S \subseteq AR$ iff $\exists Y$ s.t. $(Y, X) \in Att$ implies $\exists Z \in S$ s.t. $(Z, Y) \in Att$. Let $S \subseteq A$ be *conflict free*, i.e., there are no A, B in S such that $(A, B) \in Att$. Then, S is a *complete* extension iff $X \in S$ whenever S is acceptable with respect to S and S is a *grounded* extension, GE, iff it is a complete extension that is minimal with respect to set inclusion.

3. Formalisation of dialectical interaction: from the Lakatosian model of mathematical discourse to a formal dialogue system

In this section we express the Lakatos model of collective proof as a formal dialogue system, the Lakatos Game (LG), understood in a way of standard representation for persuasion dialogues introduced in [79] and described in Section 2.3. We assume that this type of dialectical interaction is in fact the type of persuasion, since the initial situation of conflicting opinions on whether the conjecture is true or not is sought to be resolved by the players through communication [103]. However, the nature of the interaction is not "selfish" (as in the most radical type of persuasion where a player is interested only in the situation in which s/he is winning the game), because both parties ultimately aim to collaboratively test whether the conjecture is true and whether it can be proved, no matter who of them will win (such a type of persuasion is called collaborative conflict resolution in [19]).

LG determines rules for playing a game of collaborative mathematics and in this sense it is a formal representation of informal theory introduced by Lakatos. The rules of the LG system are designed as a bridge between the spirit of the loosely defined techniques in [55] and the formality required for expressing dialogues as complete specifications that, ultimately, can be implemented. According to the standard introduced in Section 2.3, LG is specified through five types of rules: (1) locution rules which determine what types of moves players are allowed to perform during the dialogue (i.e. what are legal locutions), and, uniquely for LG, how these moves update the current mathematical *theory* in which the co-constructed proof is sited; (2) structural rules which regulate what types of responses are allowed to be given (i.e. what are legal responses); (3) commitment rules which specify a set of propositions to which players will be committed as a result of performing a given locution during the dialogue[5]; (4) termination rules which describe when the dialogue will end; and (5) outcome rules which determine what the result of a dialogue is.

The "players" in our Lakatos Games are a Proponent and an Opponent. These roles are defined relative to a given conjecture. Multiple speakers may contribute to these roles, and both the Proponent and Opponent role can be voiced by the same speaker. This allows a speaker to avoid contradictory commitments, as long as s/he does not take on both roles simultaneously. A player who takes the same role as the other speaker has to commit to the same commitments and the same strategy of proof as his predecessor(s) as the proof is done collaboratively. In other words, the game does not distinguish between participant and roles.

3.1. LG system: overview

The overall goal of the Lakatosian dialogue is to explore a mathematical theory and construct new examples, concepts, definitions, conjectures and proofs. The analysis of a fledgling proof and conjecture, via the patterns of discourse that Lakatos identified, provides a mechanism by which such a theory can grow. The Lakatos Game fulfils the definition of persuasion dialogue specified in [103,79], since the dialogue starts with a conflict of opinion about a conjecture, and aims at resolution of the conflict. In this type of dialogue, Opponent, O, of a proof disagrees with Proponent, P, and they argue about elements

[5] In formal dialogue systems, a commitment store is used to keep track of propositions to which a player is committed, i.e. to which has e publicly declared belief (though this does not mean necessarily that a player really does believe the proposition, since he can be insincere).

of the theory from which the proof is constructable. The parties aim to collaboratively solve the problem rather than winning the game at any cost, thus this type of persuasive interaction is called collaborative conflict resolution [19].

Notation. Let $b, c, d, e, f, g, k, l, m, n, r, s$ be propositional variables. For clarity in description, we consistently use specific variables to describe specific objects in Lakatos's model[6]:

- c, b: conjecture (c – current conjecture, b – new conjecture)
- l, k: lemma (l – current lemma, k – new lemma)
- m, n: counterexamples which may become monsters, if they turn out to be invalid
- d, e, f, g: definitions
- r, s: propositions used to support or contradict counterexamples

The syntax of the system allows for high level (abstract) description of the legal moves and interactions that the players can execute during the LG dialogue. In the actual game of collaborative proof creation, the variables are then instantiated by the specific statements in natural language such as in Example (1) in Section 7 and examples in Appendix A.

In the following sections we describe the locution rules (Section 3.2), structural rules (Section 3.3), and commitment, termination and outcome rules (Section 3.4) which define our formalisation of the LG System.[7]

3.2. LG system: locution rules

We specify all of the legal locutions in the LG system by the rules below, describing informally how the mathematical theory that is being co-constructed by the participants is updated by each of them; a more formal account showing how these updates are characterised in AIF and abstract argumentation frameworks then occupies much of the remainder of the paper.

Proponent and Opponent's behaviour during a collaborative proof is regulated by the LG locution rules (see below for their formal specification). In general, they allow the Proponent to create a proof (see locution rule **L1**); defend it against Opponent's attacks (counterexample, critique; see the rules **L2**–**L5**); or surrender the current conjecture (**L7.2**). Conversely, the LG locution rules allow O to: attack the proof, i.e. attack the conjecture or a lemma (**L2**); participate in the modification of a concept (see **L5.4**–**L5.5**); decide whether the counterexample is a monster or not (**L6**); or accept the current conjecture and the current lemmas (**L7.1**).

L1 Creation of proof. For creating the proof, the player P introduces a conjecture (**L1.1**) and lemma(s) (**L1.2**), and then announces the end of the informal proof by saying *ProofDone* which plays a role analogous to "■" in typeset mathematics (**L1.3**).

L2 Attack on conjecture or lemmas. For attacking the proof, Opponent has three strategies available: he can attack locally by introducing a counter m to the lemma l (*LocalCounter(m, l)*, **L2.1**); attack globally by introducing a counterexample m to the conjecture c (*GlobalCounter(m, c)*, **L2.3**); or attack in a hybrid way by introducing m which counters both c and l (*HybridCounter(m, l, c)*, **L2.2**).

L3–L5 Defending a proof. For defending the proof against Opponent's attacks, Proponent has available three strategies: he can modify the conjecture (**L3**); manipulate lemmas (**L4**); or modify the definition of a concept in the conjecture or lemmas (**L5**). In the first case, P has two alternative options of how to change the current conjecture c to a new conjecture b – he can either perform *PiecemealExclusion(b)* and directly introduce a new conjecture b (**L3.1**); or perform *StrategicWithdrawal(r, c)* and introduce r which contradicts with the current conjecture c (see **L3.2**). In the second case, Proponent acts in some sense as Opponent, i.e. he doubts that his conjecture c actually holds and uses r to demonstrate it (and then replaces it with a new conjecture b, see the structural rule **S9**).

The second strategy of defending against a counter is the manipulation of lemmas in three different ways. First, P can introduce a new lemma k to replace the current lemma l which was successfully countered by m (*LocalLemmaInc(m, l, k)*, **L4.1**). Second, Proponent can retract a lemma l which was countered by m (*HybridLemmaInc(m, l)*, **L4.2**). Third, he can add a new lemma k which will be countered by m (*GlobalLemmaInc(m, k)*, **L4.3**). The reason for proponent introducing a problematic lemma will become clear when we look at the next move that this player will perform, i.e. when we will discuss the structural rules below (see **S17** and **S18**).

[6] Note that we use variables, in the strict sense, to refer to propositions, e.g., m refers to stating that something is a counterexample. However, we also use them also as shortcuts to refer to the objects described by these statements, e.g., we use m to refer to the counterexample itself.

[7] Notice that the terminology adopted in this paper might be slightly unclear or ambiguous across different traditions and disciplines that this work builds upon. Although, in fact all Sections 3–5 talk about some formal aspect of Lakatos theory, we use the phrase "formalisation" specifically to the system introduced in Section 3 in order to reflect and emphasise the idea that LG aims to describe in the precise manner the informal, philosophical model proposed by Lakatos. Then, for Sections 4 and 5, we use terms "representation" and "evaluation" to make the reference to two traditionally distinguished key stages or areas of study of argumentation (the distinction introduced by Aristotle and then wildly used in argumentation theory).

LG Locution rules

L1 Creation of proof

1. *Conjecture(c)* asserts a conjecture *c*;
   c is added to the current theory
2. *Lemma(l)* asserts a lemma *l*;
   l is added to the current theory
3. *ProofDone* announces that a proof for the current conjecture is complete, and adds that inference, comprising the lemmas and the inference from them to the conjecture, to the theory (in place of the component parts)[8]

L2 Attack on conjecture or lemmas

1. *LocalCounter(m, l)* asserts a counterexample *m* that contradicts with a lemma *l*;
   l is removed from the current theory, and as a consequence, so is the proof as a whole from the lemmas to the conjecture
2. *HybridCounter(m, l, c)* asserts a counterexample *m* that contradicts with a conjecture *c* and a lemma *l*;
   is removed from the current theory, and as a consequence, so is the proof as a whole from the lemmas to the conjecture
3. *GlobalCounter(m, c)* asserts a counterexample *m* which supports a counter-conjecture not-*c*;
   the proof as a whole from lemmas to conjecture is removed, whilst the counterexample is added to the current theory

L3 Defence by modifying conjecture

1. *PiecemealExclusion(b)* asserts a new conjecture *b*;
   b is added to the current theory
2. *StrategicWithdrawal(r, c)* asserts *r* which contradicts with the current conjecture *c*;
   the entire proof from lemmas to conjecture is removed from the current theory

L4 Defence by manipulating lemmas

1. *LocalLemmaInc(m, l, k)* asserts a new lemma *k* that incorporates the counter *m* into an existing lemma *l* with a view to replacing *l*;
   k is added to the current theory
2. *HybridLemmaInc(m, l)* retracts the lemma *l* in response to the counter *m*;
   no update to the current theory
3. *GlobalLemmaInc(m, k)* asserts a new lemma *k* which contradicts the counterexample *m*;
   m is removed from the current theory

L5 Defence by modifying concept

1. *MonsterBar(m, c, r)* asserts *r* which contradicts the justification that the counter-conjecture not-*c* holds because of *m*;
   r is added to the theory, and as a result, the entire proof from lemmas to conjecture is reinstated
2. *MonsterAdjust(m, r)* asserts *r* which contradicts with *m*;
   r is added to the theory, *m* is removed, and as a result, the entire proof from lemmas to conjecture is reinstated
3. *PDefinition(m, r, d)* asserts the definition *d* (performed by proponent) which supports *r* used to show that *m* is not a valid counterexample;
   d is added to the theory, and *r* is replaced by the argument from *d* to *r*[8]
4. *ODefinition(m, r, d, s, e)* asserts the definition *e* (performed by opponent) which contradicts *r*, and supports *s*;
   d is removed from the theory
5. *Prefer(m, r, f, g)* prefers the definition *f* over the definition *g* in response to the claim that *m* is a monster, and the claim's justification *r*;
   the definition *f* and the argument that follows from it are added to the current theory; in case the move is opponent's (i.e. preferring proponent's definition), the original proof as a whole from lemmas to conjecture is reinstated

L6 Decision about monster

1. *MonsterAccept(m, r)* re-asserts *r* in support of the monster, *m*;
   no update to the current theory
2. *MonsterReject(m, r, d, s, c)* asserts *s* which contradicts with *r* and joins the counter *m* to supports the counter-conjecture not-*c*;
   the counterexample *c* is added whilst both the arguments from *d* to *r* and from *l* to *c* are removed from the current theory

L7 Decision about proof

1. *Accept* expresses the acceptance the current conjecture and the current lemmas; no update to the current proof
2. *Surrender* expresses surrender of the current conjecture; no update to the current proof

Finally, *P* can defend the proof by modifying the definition of a concept in the conjecture and, as a result, demonstrating that *m* is not a valid counterexample, i.e. that *m* is in fact a monster. According to this strategy Proponent introduces the proposition *r* which contradicts with: either the inference from counterexample *m* to the counter-conjecture not-*c* (*MonsterBar(m, c, r)*, see **L5.1**); or directly the counter *m* (*MonsterAdjust(m, r)*, **L5.2**). In other words, in the first case Proponent does not claim that *m* does not hold, but he points out that *m* cannot be used to infer not-*c*. In order to justify that *r* holds, *P* introduces a definition of a concept in the conjecture (*PDefinition(m, r, d)*, **L5.3**). If Opponent provides his alternative definition (*ODefinition(m, r, d, s, e)*, **L5.4**), then the players have to decide which definition (*f* or *g*) they prefer (*Prefer(m, r, f, g)*, **L5.5**). Opponent's definition move takes more variables as a content than Proponent's definition move, because *O* not only attacks *P*'s definition, but also defends his move of rejecting that the counterexample is a monster *MonsterReject(m, r, d, s, c)* (see **L6.2** described below). More specifically, *O* introduces his own definition *e* which contradicts *P*'s definition *d* and at the same time *e* supports a proposition *s* which contradicts with Proponent's attack *r* from the move *MonsterBar(m, c, r)* or *MonsterAdjust(m, r)*. Note that the main difference between the strategy of modifying the conjecture and the strategy of modifying a concept in this conjecture lies in the consequences of these moves for a theory in which the proof is conducted. The modification of the conjecture influences only the conjecture itself, while the modification of the concept influences all the conjectures in the theory which contain this concept.

L6 Decision about monster. For deciding whether the counterexample is a monster, *O* can assert *r* which *P* previously introduced to demonstrate that *m* is a monster. In other words, Opponent agrees with Proponent by repeating what Proponent said to attack the counter (*MonsterAccept(m, r)*, **L6.1**). Alternatively, *O* can reject that *m* is a monster by asserting *s* which contradicts Proponent's attack *r* and is included as a linked premise into his argument that *m* supports the counter-conjecture not-*c* (*MonsterReject(m, r, d, s, c)*, **L6.2**).

[8] This replacement is not a feature intrinsic to Lakatos' account, but is a convenience introduced here to harmonise the style of representing inference used later – nothing of significance hangs upon this way of characterising inference and entailment.

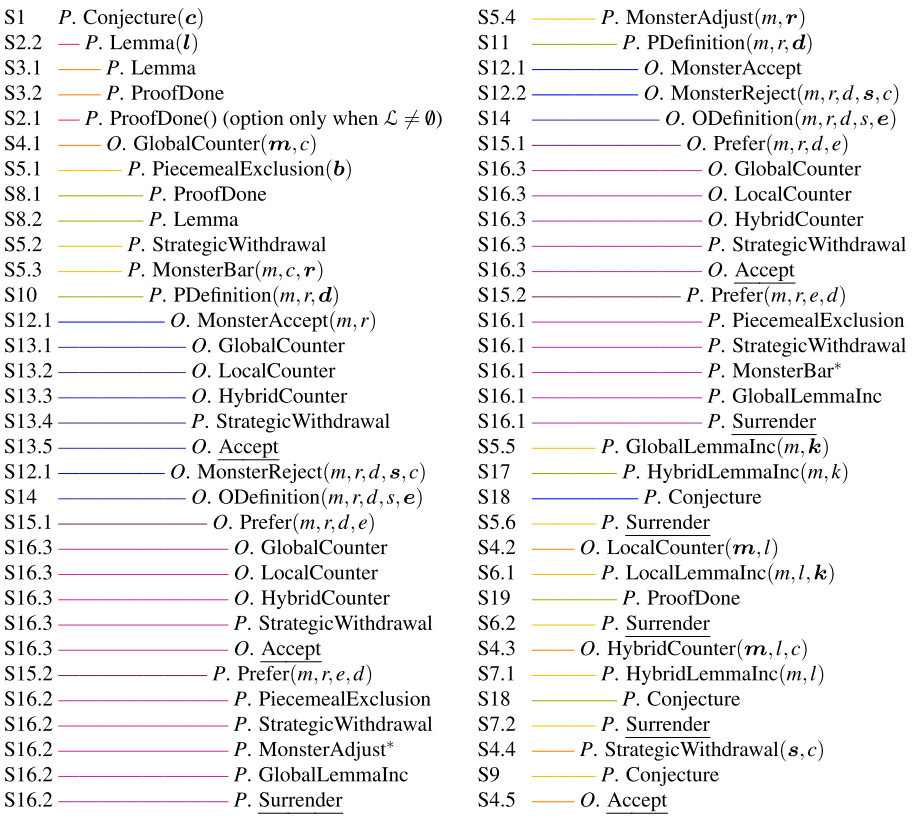


Fig. 3. Behaviour Tree for the LG system. Bold indicates where a new variable has been introduced in a given path down the tree. The names of moves without any variables represent links to positions that appear higher up in the tree. Underlining indicates moves that end the game. Note that structure under MonsterBar and MonsterAdjust is similar but not identical: the point of difference is indicated with asterisks.

L7 Decision about proof. For deciding about the acceptance of the proof, Proponent can only surrender the conjecture (*Surrender*, **L7.2**). For deciding about the acceptance of the proof, Opponent can only accept the proof (*Accept*, **L7.1**).

3.3. LG System: structural rules

The dynamics of the LG System are determined by the structural rules which describe how the players can navigate through the space of the legal locutions (see below for their formal specification). Notice that each rule determines a set of possible reposes to this move, and the players have to choose one and only one of alternatives. A different response can be chosen once the players will return to a given state. Fig. 3 is a behaviour tree that summarises the permitted follow-up structure that is detailed below. In the figure, items with one greater level of indenting are potential follow-up moves. Terms that are not terminating and that have no descendants should be understood as links that redirect the flow to nodes that appear higher up in the tree.[9]

S1–S4 Creation of proof. At the beginning of the game Proponent creates an initial, draft proof. His first move is to introduce a conjecture c (rule **S1**). At that point no lemma has been introduced yet, so he has to propose at least one lemma l according to **S2.2**. This rule is also used to prohibit Proponent from removing all lemmas and ending the proof by saying *ProofDone* (this is allowed by the rule **S2.1**). Then, Proponent can end the proof (**S3.2**); or continue adding further lemmas (**S3.1**) and then end the proof (**S3.2**). If Proponent is not introducing the initial version of the proof, but is introducing changes to the conjecture by performing *Conjecture(c)* and there is still at least one lemma in the proof, then he can either continue introducing additional lemmas (**S2.1b**), or end the proof (**S2.1a**).

After the initial version of the proof is ended, there are three possible ways of continuing the game: either Opponent attacks the proof by introducing a counterexample m to the conjecture, a lemma, or both the conjecture and a lemma (**S4.1**–**S4.3**) together; or Proponent changes his mind and introduces a new conjecture in order to prevent the proof from a potential future attack from opponent (i.e. a potential counterexample for which the old conjecture might not hold, **S4.4**); or Opponent accepts the proof (**S4.5**).

[9] A diagram of this process is also available in the form of a finite-state machine, online at http://arg.tech/lakatosFSM.

LG Structural rules

S1 A player P moves first with *Conjecture(c)*; then each player contributes a locution according to the rules **S2**–**S19** with the restriction that if a move has some propositional content, then players cannot perform this move again with the same content

S2 After *Conjecture(c)*:

1. if the lemma commitment store $\mathcal{L}$ is not empty, then P can:
   (a) end the proof: *ProofDone*
   (b) introduce a lemma l: *Lemma(l)*
2. if the lemma commitment store is empty, then P must introduce a lemma l: *Lemma(l)*

S3 After *Lemma(l)*, P can:

1. perform a sequence of locutions introducing some finite number of additional lemmas: *Lemma(k)*, ...
2. end the proof of a current conjecture by saying: *ProofDone*

S4 After *ProofDone*:

1. O can introduce a counterexample m to the conjecture c: *GlobalCounter(m, c)*
2. O can introduce a counterexample m to a lemma l: *LocalCounter(m, l)*
3. O can introduce a counterexample m to l and c: *HybridCounter(m, l, c)*
4. P can attack the current conjecture c with r: *StrategicWithdrawal(r, c)*
5. O can accept the proof: *Accept*

S5 After *GlobalCounter(m, c)*, P can reply:

1. *PiecemealExclusion(b)* introducing a new conjecture b
2. *StrategicWithdrawal(r, c)* attack the current conjecture c with r
3. *MonsterBar(m, c, r)* introducing r which contradicts with the justification that the counter-conjecture not-c holds because of the counterexample m
4. *MonsterAdjust(m, r)* introducing r which contradicts m
5. *GlobalLemmaInc(m, k)* introducing a new lemma k which contradicts m
6. *Surrender*

S6 After *LocalCounter(m, l)*, P can reply:

1. *LocalLemmaInc(m, l, k)* introducing a new lemma k which replaces l and incorporates the counter m
2. *Surrender*

S7 After *HybridCounter(m, c, l)*, P can reply:

1. *HybridLemmaInc(m, l)* retracting a lemma l which was countered by m
2. *Surrender*

S8 After *PiecemealExclusion(b)*:

1. if the lemma commitment store is not empty, then P can:
   (a) end the proof for the new conjecture b: *ProofDone*
   (b) introduce a new lemma k: *Lemma(k)*
2. if the lemma commitment store is empty, then P introduces a lemma: *Lemma(k)*

S9 After *StrategicWithdrawal(r, c)*, P introduces a new conjecture: *Conjecture(b)*

S10 After *MonsterBar(m, c, r)*, P introduces a definition d which justifies the proposition r contradicting the justification that not-c holds because of m: *PDefinition(m, r, d)*

S11 After *MonsterAdjust(m, r)*, P introduces a definition d which justifies the proposition r contradicting the counter m: *PDefinition(m, r, d)*

S12 After *PDefinition(m, r, d)*, O can reply:

1. *MonsterAccept(m, r)* agreeing that m is a monster because of r
2. *MonsterReject(m, r, d, s, c)* disagreeing that m is a monster with respect to c by asserting s which contradicts with r justified by d

S13 After *MonsterAccept(m, r)*:

1. O can introduce a counterexample n to the conjecture c: *GlobalCounter(n, c)*
2. O can introduce a counterexample n to a lemma l: *LocalCounter(n, l)*
3. O can introduce a counterexample n to c and l: *HybridCounter(n, l, c)*
4. P can attack the current conjecture c with s: *StrategicWithdrawal(s, c)*
5. O can accept the proof: *Accept*

S14 After *MonsterReject(m, r, d, s, c)*, O can introduce a definition e which contradicts with Proponent's definition d and justifies the proposition s asserted in the move *MonsterReject*: *ODefinition(m, r, d, s, e)*

S15 After *ODefinition(m, r, d, s, e)*:

1. O can introduce a preference of Proponent's definition d over her own definition e: *Prefer(m, r, d, e)*
2. P can introduce a preference of Opponent's definition e over her own d: *Prefer(m, r, e, d)*

S16 After *Prefer(m, r, f, g)*:

1. if *Prefer(m, r, f, g)* is performed by P after *MonsterAdjust(m, r)*, then P can reply:
   (a) *PiecemealExclusion(b)* introducing a new conjecture b
   (b) *StrategicWithdrawal(s, c)* attacking the current conjecture c with s
   (c) *MonsterBar(m, c, s)* introducing s which contradicts the justification that the counter-conjecture not-c holds because of the counter m
   (d) *GlobalLemmaInc(m, k)* introducing a new lemma k which contradicts m
   (e) *Surrender*
2. if *Prefer(m, r, f, g)* is performed by P after *MonsterBar(m, c, r)*, then P can reply:
   (a) *PiecemealExclusion(b)* introducing a new conjecture b
   (b) *StrategicWithdrawal(s, c)* attacking the current conjecture c with s
   (c) *MonsterAdjust(m, s)* introducing s which contradicts the counter m
   (d) *GlobalLemmaInc(m, k)* introducing a new lemma k which contradicts m
   (e) *Surrender*
3. if *Prefer(m, r, f, g)* is performed by O, then:
   (a) O can introduce a counterexample n to the conjecture c: *GlobalCounter(n, c)*
   (b) O can introduce a counterexample n to a lemma l: *LocalCounter(n, l)*
   (c) O can introduce a counterexample n to c and l: *HybridCounter(n, l, c)*
   (d) P can attack the current conjecture c with s: *StrategicWithdrawal(s, c)*
   (e) O can accept the proof: *Accept*

S17 After *GlobalLemmaInc(m, k)*, P retracts the lemma k which was countered by m: *HybridLemmaInc(m, k)*

S18 After *HybridLemmaInc(m, l)*, P replies *Conjecture(b)*

S19 After *LocalLemmaInc(m, l, k)*, P replies *ProofDone*

S5–S7 Attack on conjecture or lemmas. Opponent's attack on the conjecture, via *GlobalCounter(m, c)*, generates the largest number of possible alternative paths through the Lakatos game (**S5**, see also Fig. 3). First, P can immediately give up, i.e., he can become convinced of Opponent's critique and decide to surrender the proof (**S5.6**). Another way to respond is to only

partly agree with *O* and modify the conjecture so that the attack fails (**S5.1–S5.2**). Proponent can introduce the changes directly either by piecemeal exclusion, i.e., introducing a new conjecture *b* (**S5.1**), or by strategically withdrawing (**S5.2**), i.e., expressing doubts about the current conjecture *c* first.

Opponent's attack on a lemma, *LocalCounter(m, l)*, and an attack on both a lemma and the conjecture, *HybridCounter(m, c, l)*, generate much simpler histories for LG dialogue games. In the first case, Proponent can either replace a lemma with a new one which incorporates Opponent's critique (*LocalLemmaInc(m, l, k)*, **S6.1**) and then he announces that the new proof (with the new lemma) for the old conjecture is completed by saying *ProofDone* (**S19**); or *P* can agree with the critique and surrender the proof (**S6.2**). In the case of *HybridCounter* attack, *P* can either retract a lemma which was countered by *m* (*HybridLemmaInc(m, l)*, **S7.1**) and then incorporate the critique into a new conjecture *b* (*Conjecture(b)*, **S18**); or *P* agrees with the critique and surrenders the proof (**S7.2**).

S8–S9 Defence by modifying conjecture. The difference between these piecemeal exclusion and strategic withdrawal is that *PiecemealExclusion(b)* has to be always triggered by Opponent's attack, while *StrategicWithdrawal(r, c)* can be performed at any point of the game when Proponent changes his mind about the acceptance of the conjecture, even immediately after ending the initial proof (see **S4.4**). After *StrategicWithdrawal(r, c)*, *P* introduces a new conjecture *b* (**S9**).[10] In both cases, after introducing the new conjecture *b*, *P* can propose some additional lemmas or end the new proof in the same way as he did at the beginning of the game (see **S8** for excluding piecemeal and **S2** for strategically withdrawing).

S10–S16 Defence by modifying concept and decision about monster. The final type of response to *GlobalCounter(m, c)* is for Proponent to entirely disagree with Opponent's global counter. In other words, *P* tries to show that *m* is not a valid counterexample (i.e. that *m* is a monster) either indirectly by introducing *r* which attacks the inference between *m* and the counter-conjecture not-*c* (*MonsterBar(m, c, r)*, see **S5.3**); or by introducing *r* which directly attacks the counter *m* itself (*MonsterAdjust(m, r)*, see **S5.4**). Both of these moves require Proponent to formulate a new definition *d* for a concept in the current conjecture which justifies the attack *r* (**S10** and **S11**). Next, Opponent can accept the definition and as a result agree that *m* is the monster (*MonsterAccept(m, r)*, **S12.1**); or reject it and assert *s* which contradicts with *r* (*MonsterReject(m, r, d, s, c)*, **S12.2**). If *O* accepts that *m* is a monster, then he can try to attack using a different counterexample to the conjecture, a lemma or both (**S13.1–S13.3**); or he can become convinced and decide to accept the proof (**S13.5**); alternatively Proponent can modify the conjecture (**S13.4**). If *O* rejects that *m* is a monster, then he has to propose his own definition *e* of this concept which supports his claim *s* which is in contradiction to Proponent's claim *r* (**S14**). Then, one of the players has to decide which definition is preferred: either Opponent prefers Proponent's definition over his own (**S15.1**), or Proponent prefers Opponent's definition (**S15.2**).[11] If Proponent expresses the preference, then *P* can next: change the conjecture through piecemeal exclusion (**S16.1.a** and **S16.2.a**) or strategic withdrawal (**S16.1.b** and **S16.2.b**); try to show again that *m* is not a valid counterexample by monster-adjusting (if previously he did monster-barring (**S16.1.c**)), or by monster-barring (if previously he did monster-adjusting (**S16.2.c**)); introduce a new lemma that contradicts *m* (**S16.1.d** and **S16.2.d**); or surrender the proof (**S16.1.e** and **S16.2.e**). If Opponent expresses the preference, then: *O* can attack again by introducing another counter to the conjecture (**S16.3.a**), or a lemma (**S16.3.b**), or to both (**S16.3.c**); *P* can change the conjecture by strategically withdrawing (**S16.3.d**); or, finally, *O* can become convinced and decide to accept the proof (**S16.3.e**).

S17–S19 Defence by manipulating lemmas. Proponent can also introduce the changes to the conjecture indirectly, i.e., by firstly modifying a lemma and then modifying the conjecture. More specifically, first *P* performs *GlobalLemmaInc(m, k)* which introduces a new lemma *k* contradicting the counter *m* (**S5.5**). This lemma reveals the hidden assumption that needs to hold in order for the conjecture to hold too. Then, *P* continues with hybrid lemma incorporation which retracts the recently introduced lemma *k* (**S17**) and then incorporates the hidden assumption into a new conjecture *b* (**S18**).

3.4. LG System: commitment rules, termination rules and outcome rules

The effects of the players' interactions are determined by three final types of rules: the commitment rules determine the effects that these interactions have on commitment stores of the players; termination rules define when the dialogue ends; and outcome rules describe who won the dialogue. See below for their formal specification for the LG System.

[10] Notice that the rule for *PiecemealExclusion(b)* in **S8** repeats the specification for *Conjecture(c)* in **S2**, which means that in the first case Proponent again introduces a new conjecture directly as a response to Opponent's attack rather than doing so via a *Conjecture(b)* move.

[11] Notice that it is prohibited by the LG rules for both players to prefer each other's definitions, or for neither of them to prefer the other's definition. According to the rule **S15**, the players have to choose one and only one of the *Preference* moves – either Proponent will chose to prefer Opponent's definition or vice versa. As soon as one of them does the *Preference* move, the set of available moves change (described now by the rule **S16**) and the other player cannot perform the *Preference* move anymore. The protocol determines what should be the next move in the game to match the idea of collaborative mathematics proposed in the Lakatos model, but it does not determine what the players should say. In other words, according to the model the players should resolve between themselves which of alternative definitions is better in order to continue collaborative proof.

LG Commitment rules

C1 Conjecture commitment store

1. After *P* performs *Conjecture(b)* or *PiecemealExclusion(b)*, the conjecture store is emptied and the propositional content *b* is included in the store

C2 Lemma commitment store

1. After *P* performs *Lemma(k)* or *GlobalLemmaInc(m, k)*, the content *k* is included in the lemma commitment store
2. After *P* performs *LocalLemmaInc(m, l, k)*, *l* is removed from the store and *k* is added
3. After *P* performs *HybridLemmaInc(m, l)*, *l* is removed from the store

LG Termination rules

T1 A dialogue terminates if either *Accept* or *Surrender* is performed

LG Outcome rules

O1 Proponent wins, if a dialogue terminates with *Accept*

O2 Opponent wins, if a dialogue terminates with *Surrender*

In the LG system for collaborative mathematics, there are two types of commitment stores: the store for the conjecture (see **C1**) and the store for lemmas (**C2**). These stores keep track of the currently posited lemmas and conjecture. Most commitment updates are forced by Opponent's attacks (i.e., when Proponent is not able to defend against an attack, then he is forced to change the current proof). Similarly, the information about the other propositions used by players in counterexamples, definitions and so on, is not stored at all, since they do not contribute directly to the proof – they only influence its shape during the game.

Proponent can add a conjecture *b* into the conjecture commitment store by performing either *Conjecture(b)* or *PiecemealExclusion(b)* (**C1**). This type of store has to consist of only one proposition at any time of the game (i.e., it has to consist of the current conjecture), therefore both of these moves first empty the store to remove the old conjecture (if there is one), and then adds a new conjecture *b*. Further, *P* can add lemmas to, and remove them from, the lemma commitment store in three different ways: (i) adding a new lemma *k* either by performing *Lemma(k)* or *GlobalLemmaInc(m, k)* (**C2.1**); (ii) replacing lemma *l* with a new lemma *k* by performing *LocalLemmaInc(m, l, k)* (**C2.2**); or (iii) removing lemma *l* from the store by performing *HybridLemmaInc(m, l)* (**C2.3**).

The dialogue terminates when Opponent performs *Accept*, or when Proponent performs *Surrender* (see (**T1**)). Proponent wins if Opponent accepts the proof (**O1**); and Opponent wins if Proponent surrenders the proof (**O2**).[12] Note that the phrase "win" is not used here to suggest that there is a competitive character to LG dialogues, since we assume, like Lakatos, that they are modelling a collaborative process. Our aim is just to stay also close to the standard terminology used in formal dialogue systems. As a result, Proponent's "winning" should be interpreted as the acceptance of the current conjecture and the current lemmas, and Opponent's "winning" should be interpreted as surrender of the current conjecture.

4. Graph-based representation: from a formal dialogue system to Argument Interchange Format structures

The majority of research in the philosophy of dialogue focuses on normative, idealised models of interaction; only quite recently has there been an empirical turn that aims to connect such models with discursive practice. Normative games like DC [62] aim to make explicit features of dialogues which are latent in natural interactions (in the case of DC, cumulativity, whereby retraction is prohibited). Descriptive games such as PPD [103] aim to express natural interaction with more formal apparatus, thereby allowing dialogues to be assessed and guided (for comparative analysis of DC, PPD and many other games, see [105]). The Lakatos Game is designed with both normative and descriptive flavours, providing scaffolding and guidance for practical interactions, as well as tools that can be used to interpret and assess discourse. Crucially though, Lakatos games are not aimed, like DC and PPD, at establishing a common understanding, or a maieutic exploration of a space; they are instead aimed quite specifically at generating a proof, or, more specifically, at yielding a theory from which a proof can be extracted.

As a consequence of this normative-descriptive balance, the Lakatos Game provides a good test case for AIF, which similarly aims to handle both analysis of linguistic material as well as representation and evaluation of the norms of discourse context [25]. For although AIF structures are designed to handle structures of argument, their closest counterparts in formal logical systems are not formal theories or sets of propositions, but proofs. If the description of LG is adequate, therefore, it should yield AIF structures which can automatically be mapped to such proofs.

There is, however, a challenge. AIF is used as infrastructure for an interconnected web of debates and arguments that allows navigation through different modes, domains, types of argument, with the ability to extend, critique and adumbrate them via a range of systems and tools. This infrastructure – collectively referred to as the Argument Web [16] – is founded upon several assumptions. One cornerstone is the assumption of cumulativity: once an argument has been committed to the

[12] Notice that Proponent cannot perform *Surrender* when he is winning, because the structural rules allow him to do it only when he does not choose to defend the proof as a response to an attack. For example, in the case of the attack of *GlobalCounter(m, c)* (described by the series of the rules **S5–S7**), Proponent can immediately give up without a defence (**S5.6**), otherwise he chooses to recover the status of the proof and the defence makes him being in the winning position until the next attack. In a similar way, the structural rules allows Opponent to perform *Accept* only when he decided to not continue attacks, conceding as the result that Proponent managed to successfully defend his proof. For example, Opponent cannot reply *Surrender* to *LocalCounter*, because it is prohibited by the rule **S6**. After *LocalCounter*, the turn belongs to Proponent and he can either give up (conceding as the result that Opponent's attack is successful) or defend his position with *LocalLemmaInc(m, l, k)*.

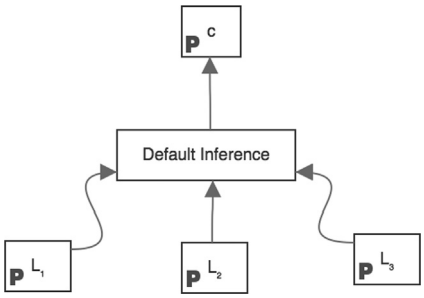


Fig. 4. Lemmas, conjecture and proof.

Argument Web, it is there in perpetuity. Although individual arguers may change their position or retract their arguments, the arguments themselves must remain available so that others can refer to them, build upon them and revise them.

But if the Argument Web [16] is to capture the emerging proofs as they are developed, extended, redefined and reframed by a Lakatos game, how can the intrinsic nonmonotonicity be reflected by cumulative infrastructure? Our goal is to ensure that the semantics of the moves in the game, as they are defined in terms of AIF updates, should yield argument structures with a specific set of properties. These argument structures should be submittable to a simple, algorithmic procedure which will yield precisely the theory that represents the current shared understanding at any point in the game. To deliver this, the effects that each move has on AIF structures need to be carefully defined, then the argumentation semantics computation can be shown to yield exactly the correct subset corresponding to proof structures.

4.1. AIF structure update: AIF operations of LG *locutions*

For Proponent's $Conjecture(c)$ and $Lemma(l_i)$ moves, all that needs to be done is to add c and the l_i to the AIF graph. The *Proof Done* move is used to establish the inferential connection between them (in AIF terms, it adds an RA-node with incoming edges from each of the l_i and an outgoing edge to the c). In other words, the Lakatosian concept of a proof is being modelled as the inference or argument from lemmas to conjecture. The inference is established by using proponent's commitment stores, connecting all those propositions in the Lemma Commitment Store to that in the Conjecture Commitment Store. Fig. 4 visualises the AIF structure available after proponent has offered a putative proof for a conjecture, c, based on lemmas L_1, L_2 and L_3. The visualisation adopts a common convention of AIF, with both propositions and relations between them (of inference, conflict, and later, preference and restatement) expressed in boxes, and with directionality (e.g. of inference), as defined in [25], indicated by directed edges. For convenience, each proposition is also indicated as belonging to either proponent (P) or opponent (O).

There are three types of counter that Opponent can offer in response to such an argument: countering the conjecture, countering one of the lemmas, or countering both conjecture and lemma simultaneously. These are referred to as *GlobalCounter*, *LocalCounter* and *HybridCounter*, respectively. All three types of counter introduce conflict into the AIF structure:

- Opponent's $GlobalCounter(m, c)$ performs a total of four updates to the AIF structure. First, it adds m to the graph and uses it as a premise in a new argument. The conclusion of that argument is the negation of the conjecture, c. The negation is glossed as "It is not the case that c", and the second step is to add the new element to the AIF. The relationship between c and not-c is clearly one of (symmetrical) conflict, so the third step is to add new CA nodes between c and not-c. Finally, the AIF is also updated to reflect the argument from m to not-c with a new RA between them.
- Opponent's $LocalCounter(m, l)$ simply introduces the counterexample, m and the symmetrical conflict (i.e. a pair of CA nodes) between that and the lemma, l.
- Opponent's $HybridCounter(m, l, c)$ performs the same update as *LocalCounter*, with additional conflict added between m and c. We might expect *HybridCounter* to cause updates that constitute the union of the effects of *GlobalCounter* and *LocalCounter*, introducing an attack with the lemma and a support for a counter-conjecture. Instead though, *HybridCounter* uses a simpler characterisation of attack between m and c, introducing a CA directly, rather than adding an argument for the negation of the conjecture. The reason for this discrepancy is that the negation is required for the action of future moves permitted by LG after *GlobalCounter* (*viz. MonsterBar* and *MonsterAdjust*), whilst these moves are not legal following *HybridCounter*, so the characterisation can be simpler.

These three updates for handling opponent counters are visualised in Fig. 5. Proponent's eight possible substantive responses (excluding *Surrender*) to a counter then update the AIF graph further:

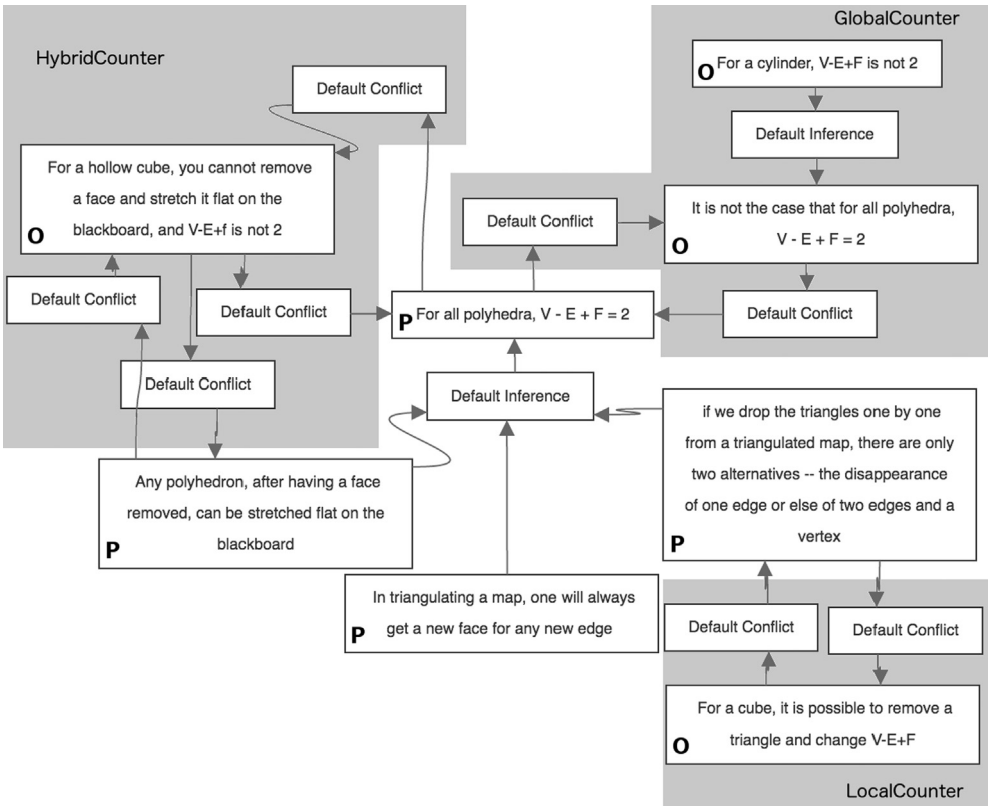


Fig. 5. Global, Local and Hybrid counters.

- *PiecemealExclusion* allows for a revised conjecture, b, to be asserted, and demands a new *Proof Done* move to create the inference from all of the current lemmas to the new conjecture. Note that this creates a new inference (i.e. a new RA-node) – the old one remains in the AIF graph.
- *StrategicWithdrawal* allows Proponent to alter the conjecture, first through introducing a reason, r, for rejecting the current conjecture, then, through the obligatory subsequent *Conjecture* move, proposing a new conjecture. The AIF is updated to reflect the addition of the new piece of information, r, plus the symmetrical conflict between that and the old conjecture, c.
- *MonsterBar* allows Proponent to challenge the counterexample offered by Opponent by changing the definition of terms upon which it depends. The *MonsterBar* move itself updates the AIF in two steps. First, it adds a reason, r, for rejecting the counterexample and treating it as a monster. Then a conflict is established from r to the RA-node constituting Opponent's extant argument from counterexample, m, to counter-conjecture, not-c. That is, r is treated as an undercutter. These results are visualised in Fig. 9.
- *MonsterReject*-ing allows Opponent the same freedom as Proponent in offering a reason for thinking the counterexample does, indeed hold. This reason s is naturally in conflict with Proponent's reason that the counterexample does not hold, so the AIF updates are first to add s and to link it as a further premise in the argument from the example, m, to the counter-conjecture, not-c (that is, to connect s to the RA-node between m and not-c). Then the symmetrical conflict between s and r is also added. These updates are also visualised in Fig. 9.
- *MonsterAdjust*-ing follows the same pattern as *MonsterBar*-ing, except that the counterexample is attacked directly by redefinition. That is, *MonsterAdjust* introduces a counter, r, and the symmetrical conflict between r and m. An intuition remains, however, for *MonsterAdjust*: that the counter to r somehow trumps the monster. To capture this intuition, we further add a preference(PA) that allows r to defeat m. Both *MonsterBar* and *MonsterAdjust* then open a phase of the dialogue in which definitions are introduced (see below). The results of this move are visualised in Fig. 10.
- *HybridLemmaInc*-orporation, following a *HybridCounter* paves the way for the introduction of a revised conjecture, c' which incorporates one lemma, l_i. The update, however is performed by a subsequent *Conjecture* move.
- *GlobalLemmaInc*-orporation makes explicit a hidden assumption by adding it as a new lemma, k. It is also responsible for introducing the conflict between the counterexample, m (identified in the preceding *GlobalCounter*) and this new lemma. It then forces a *HybridLemmaInc*-orporation so that k is incorporated into a new conjecture and thence proof.
- Finally, *LocalLemmaInc*-orporation works in a similar way to *GlobalLemmaInc*-orporation, except that it introduces a revised lemma k before adding a new inference connecting all lemmas and (unchanged) conjecture through a mandatory

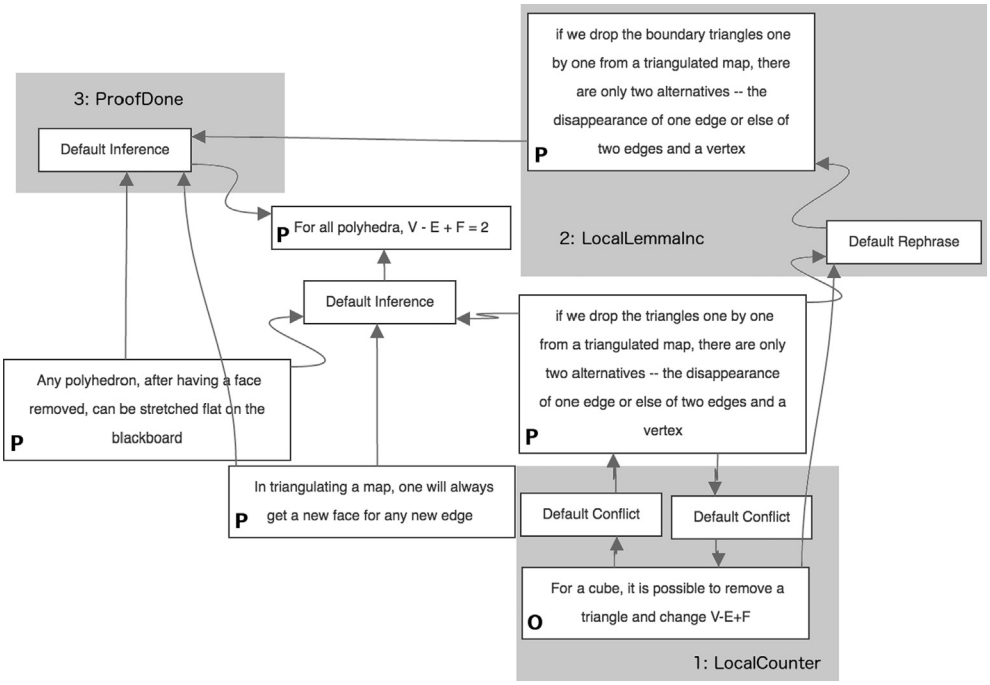


Fig. 6. Local lemma incorporation.

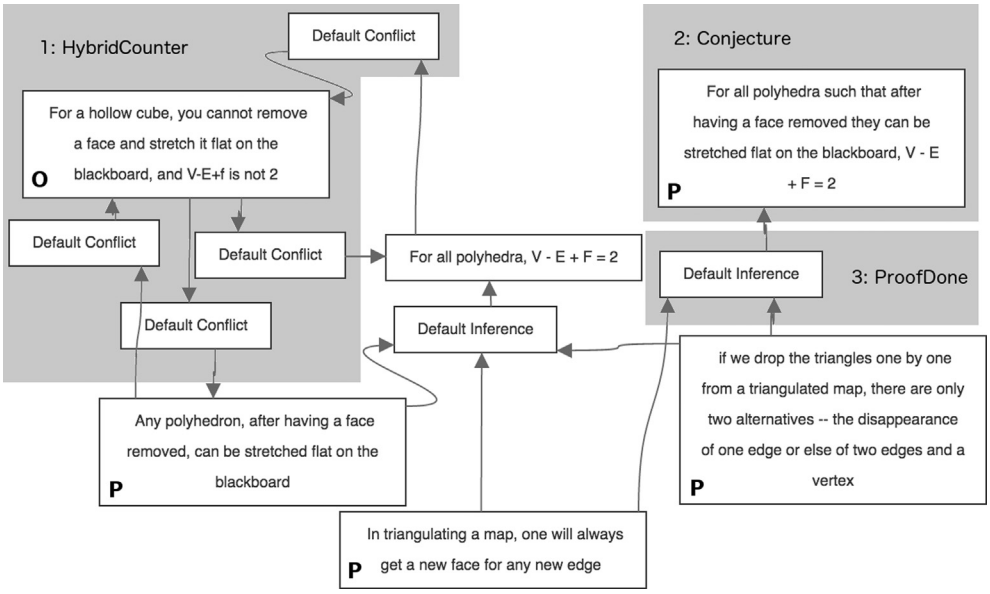


Fig. 7. Hybrid lemma incorporation.

Proof Done move. In addition, AIF provides the ability to represent non-inferential relations between propositions, including paraphrastic relationships such as restatement. We can capture the 'incorporates' relationship in this way as an MA node, but this has no impact on the argument frameworks – structured and abstract – that are created, functioning currently only to make the linguistic relationship. The AIF updates effected by the three types of lemma incorporation are visualised in Figs. 6–8.

The moves of *MonsterBar* and *MonsterReject* can lead to further exchanges regarding the definitions upon which the counterexample depends, each of which provides further updates to the AIF structure:

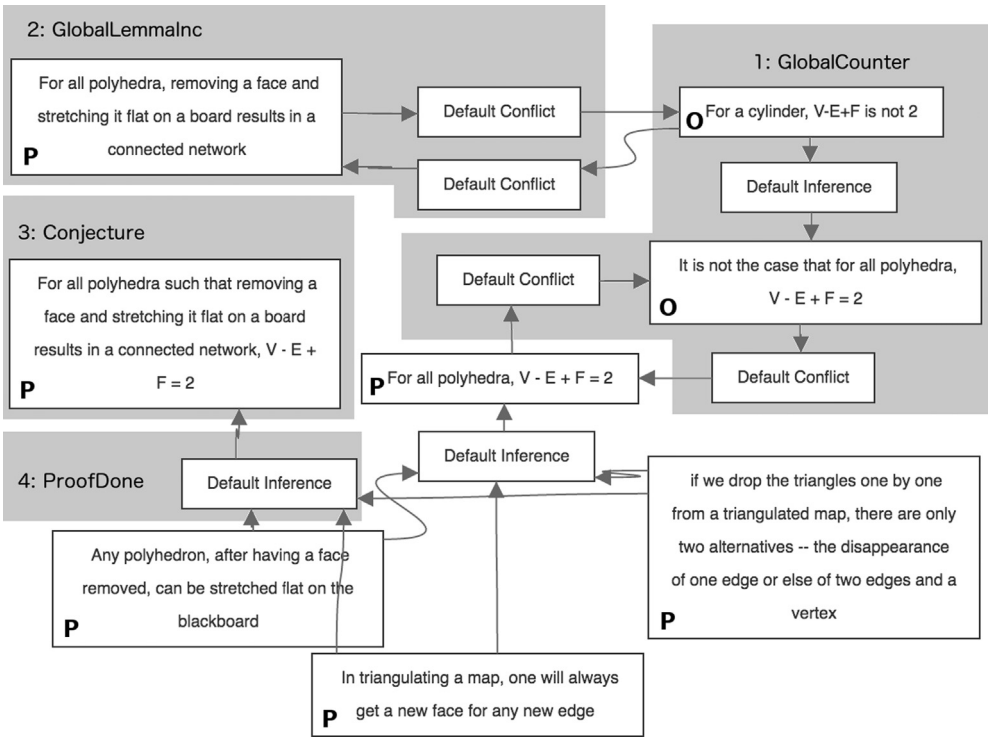


Fig. 8. Global lemma incorporation.

- *PDefinition* allows Proponent to offer a definition, *d* captured as a proposition in the AIF structure, which is then linked through an inference to the reason, *r* for barring or adjusting a monster (from a *MonsterBar* or *MonsterAdjust* move). So *PDefinition* adds both the proposition *d* and the RA from that to *r*. Proponent's definition can be *MonsterAccept*-ed or *MonsterReject*-ed. (The idea of definitions being used as the bases for arguments is one that is familiar in the philosophy of argument – see, e.g., [1].)
- *MonsterAccept*. In the former case, there is no further AIF update because Opponent is simply reiterating Proponent's position. In the latter case, Opponent can *MonsterReject*, which has already been treated, above.
- *ODefinition* allows Opponent to offer an alternative definition, which requires update to the AIF structure not only to include the new definition, *e*, but also (i) the conflict with Proponent's definition and (ii) the inference from *e* to Opponent's reason, *s*, for thinking the counterexample holds.
- *Prefer* allows resolution between competing definition by introducing a relation of preference between two definitions. The move thus introduces a PA node between definitions *f* and *g* (the variables are thus named to emphasise that *either* Proponent's definition *d* *or* Opponent's definition *e* may be preferred).

The effects of all of these moves are shown in Figs. 9 and 10, which show the structures built up through the monster barring and monster adjusting pathways, respectively. Other moves described in the previous section impact the dialogical dynamics of the game (i.e. of what can be said, and of what commitments exist in stores) but do not update AIF structures beyond that.

5. Evaluation using structured and abstract argumentation frameworks: from AIF structures to abstract argumentation frameworks

AIF structures are one way to handle 'structured argumentation.' Other approaches, such as that provided by *ASPIC*$^+$ [80], have been shown to be compatible, in that it is possible (under certain assumptions[13]) to translate from AIF to *ASPIC*$^+$ [16]. Prakken has further shown (*ibid.*) that *ASPIC*$^+$ structures can be used to induce abstract argumentation frameworks which can make use of the wide range of existing argumentation semantics for computing acceptability.

[13] These assumptions serve to exclude boundary cases from the AIF to ensure (i) that arguments terminate with I nodes; (ii) that RA nodes always move from one or more premises to exactly one conclusion; and (iii) that PA and CA nodes always connect exactly one incoming with exactly one outgoing node, and never serve themselves as the incoming or outgoing nodes of other S nodes – i.e. never serve as premises or conclusions, conflicting or conflicted elements, preferred or dispreferred elements. None of the AIF updates described in Section 4 introduce violations to these assumptions.

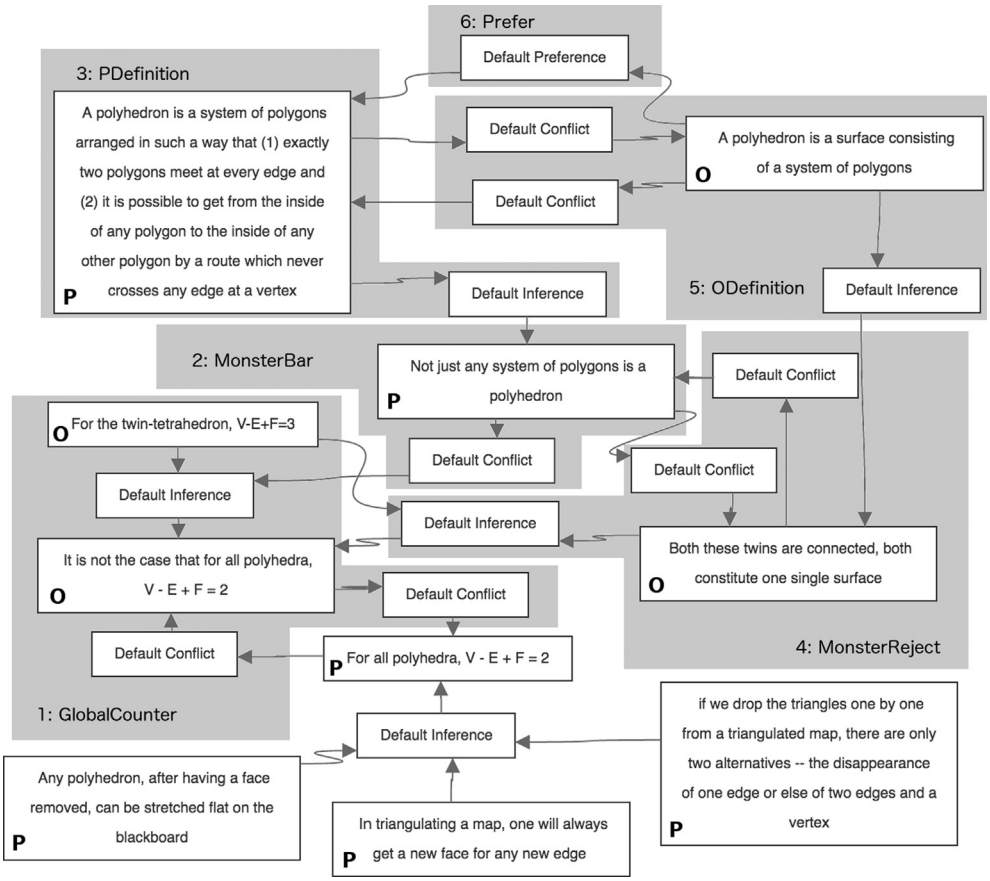


Fig. 9. Monster barring and monster rejecting, with alternative definitions and preferences between them.

The approach described in this section shows how as a Lakatos Game is executed, AIF structures are created which, when translated to *ASPIC*$^+$, produce abstract argumentation frameworks which under grounded semantics have as acceptable arguments all and only those elements which correspond to the mathematical theory accepted by the participants at a given stage of the game. Though other single extension semantics could be employed, it turns out that the strong, sceptical interpretation provided by grounded semantics deliver precisely the results required,as will be shown in this and the following sections. Our strategy is inductive, so we aim to show that each move updates the AIF graph in such a way that the grounded extension of the induced framework corresponds to the definitions of how the mathematical theory is evolving given by the LG locution rules in Section 3.2.

We describe updates in three steps. First, showing how an *ASPIC*$^+$ argumentation theory for the Lakatos Game, $AT_{LG} = ((\mathcal{L}_{LG}, \mathcal{R}_{LG}, n), \mathcal{K})$, is updated by virtue of the relationship between AIF and *ASPIC*$^+$ [16] (we adopt a convention whereby the new, post-locution argumentation theory and its constituents are referred to with a prime ($'$), whilst the original, pre-locution version is unadorned). Second, by virtue of the relationship between *ASPIC*$^+$ and abstract argumentation, we show updates to an abstract framework for the Lakatos Game LG, $AF_{LG} = \langle AR_{LG}, attacks_{LG}\rangle$, comprising a set of abstract arguments AR_{LG} and a relation of attacks $attacks_{LG}$ over them following Dung [32]. Finally, the updates to the grounded extension of the argumentation framework, GE_{LG}, are computed and compared with the status of the theory defined by the locution rules of LG. Where abstract arguments correspond to single propositions in the structured argumentation, they are labelled the same for convenience (i.e. a proposition p in the structured argumentation will correspond to an abstract argument also labelled p). Where we are interested in abstract arguments that correspond to a complex in the structured argumentation, we name them uniquely and explicitly describe their composition (so, for example, a structured argument $[p$ so $q]$ might correspond to an abstract argument α).

For clarity in presentation, the locutions are divided into three categories: those that add material to the theory (*constructive locutions*), those that attack material already in the theory (*critical locutions*) and those that make no explicit update to the material in the theory (*neutral locutions*).

Fig. 10. Monster adjusting with alternative definitions and preferences between them.

5.1. Constructive locutions

For *Conjecture(c)* and *Lemma(l)*, the update to AT_{LG} is straightforward in that $\mathcal{K}$ is updated to include them (i.e. $\mathcal{K}' = \mathcal{K} \cup \{c\}$ and $\mathcal{K}' = \mathcal{K} \cup \{l\}$, respectively[14]).

Next, we must exploit the fact that the system is constrained to prohibit repetition of identical moves (i.e. moves of identical type with identical variable instantiations, but irrespective of speaker). So long as these moves have not come before,[15] and therefore the contents, c and l, have not previously been introduced, the AIF will be updated to include c or l, and there are guaranteed to be no extant conflicts with either c or l (since conflicts can only be introduced by the *GlobalCounter*, *LocalCounter*, *HybridCounter* or $StrategicWithdrawal$ moves which can only follow earlier *Conjecture* or *Lemma* moves). Thus the new abstract framework after a *Conjecture(c)* move, AF'_{LG}, is expanded from the old thus: $AR'_{LG} = AR_{LG} \cup \{c\}$ and $attacks'_{LG} = attacks_{LG}$ (and similarly for *Lemma(l)*). With no conflicts, abstract arguments corresponding to c

[14] Specifically this is an update to $\mathcal{K}_p$, the subset comprising ordinary premises. Updates for LG make no use of the $\mathcal{K}_n$ subset comprising axioms.

[15] With the further commonsense assumption that within a single dialogue, a conjecture is not usable as a lemma, definition, or counterexample, and vice versa).

and l are guaranteed to have no attacking arguments, and therefore to be in the grounded extension. That is, $\nexists x$ s.t. $(x, c) \in attacks'$or$(x, l) \in attacks'$.

For *PiecemealExclusion(b)*, the revised conjecture, b is added to the AIF, and consequently $\mathcal{K}' = \mathcal{K} \cup \{b\}$. Under the same restrictions and assumptions as for *Conjecture(c)*, above, the corresponding abstract argument b is guaranteed to be in the grounded extension. The old conjecture that has been replaced is guaranteed to be excluded from the grounded extension because of the conflict introduced by the *GlobalCounter* or *Prefer*, the only moves that can legally precede *PiecemealExclusion*. Thus, $AR'_{LG} = AR_{LG} \cup \{b\}$ and $attacks'_{LG} = attacks_{LG}$ (since the necessary update to $attacks_{LG}$ has been carried out by other moves).

For *StrategicWithdrawal(r, c)*, the revised conjecture is added to the AIF along with its conflict with the old conjecture c. (Although from a protocol perspective this is a seemingly counter-strategic move for Proponent to make, it is in fact quite common in real exchanges – see, for example, Section A.1, turn 1.) In the *ASPIC*$^+$ argumentation theory, $\mathcal{K}' = \mathcal{K} \cup \{r\}$ and $\overline{c}' = \overline{c} \cup \{r\}$. In the induced abstract framework, the conflict results in new attacks between r and the abstract argument representing the proof, ω, i.e. $[l_i$ so $c]$. This does not remove the previous conjecture or the extant proof in its support, but suspends it, knocking it out of the grounded extension, whilst focus shifts to the revised, narrower conjecture. For *StrategicWithdrawal(r, c)*, the update is thus $AR'_{LG} = AR_{LG} \cup \{r\}$ and $attacks'_{LG} = attacks_{LG} \cup \{(r, \omega), (c, \omega)\}$.

For *MonsterBar*, the AIF is updated with a reason, r, which is used as an undercutter in the argument from counterexample to counter-conjecture. Clearly, $\mathcal{K}' = \mathcal{K} \cup \{r\}$ and in addition, $\overline{m \Rightarrow \text{not-}c}' = \overline{m \Rightarrow \text{not-}c} \cup \{r\}$. The reason, r, is guaranteed to be in the grounded extension since there could be no attackers; as a result, the status of the abstract argument corresponding to the inference from counterexample, m, to counterconjecture, not-c, is guaranteed to be out (since it is attacked by the abstract argument corresponding to r). With this argument out, if there are no other pending counterexamples (and the dialogue protocol constrains the discussion to handle just one at a time), the conjecture will be in the grounded extension. Thus, *MonsterBar(m, c, r)* updates $AR'_{LG} = AR_{LG} \cup \{r\}$ and $attacks'_{LG} = attacks_{LG} \cup \{(r, \alpha)\}$ where α is the abstract argument corresponding to the structured argumentation complex $[m$ so not-$c]$.

For *MonsterAdjust*, a direct counter, r, to the counterexample m, is introduced along with the symmetrical conflict between them, but also a preference to ensure r defeats m. For *ASPIC*$^+$, this means that $\mathcal{K}' = \mathcal{K} \cup \{r\}$, $\overline{m}' = \overline{m} \cup \{r\}$ and $\overline{r}' = \overline{r} \setminus \{m\}$ (this latter is the result of the AIF PA node). As with *MonsterBar*-ing, if there are no other pending counterexamples (and the dialogue protocol constrains the discussion to handle just one at a time), the conjecture will be in the grounded extension. Thus, for *MonsterAdjust(m, r)*, the associated update is $AR'_{LG} = AR_{LG} \cup \{r\}$ and $attacks'_{LG} = attacks_{LG} \cup \{(r, m), (r, \alpha)\}$, where α is the abstract argument corresponding to the structured argumentation complex $[m$ so not-$c]$.

GlobalLemmaInc, makes explicit the hidden assumption (as described by Lakatos) as a lemma (the subsequent incorporation of the lemma into the conjecture is handled by *HybridLemmaInc*). The new lemma is introduced and added in to AIF and thence the *ASPIC*$^+$ argumentation theory thus: $\mathcal{K}' = \mathcal{K} \cup \{k\}$, $\overline{m}' = \overline{m} \cup \{k\}$ and $\overline{k}' = \overline{k} \cup \{m\}$. At the abstract level, the argument corresponding to this lemma is guaranteed to be in the grounded extension. That is, for *GlobalLemmaInc(m, k)*, the update is $AR'_{LG} = AR_{LG} \cup \{k\}$ and $attacks'_{LG} = attacks_{LG} \cup \{(k, \alpha), (m, k), (k, m)\}$, where α is the abstract argument corresponding to the structured argumentation complex $[m$ so not-$c]$.

For *LocalLemmaInc(m, l, k)*, the new lemma k is added and guaranteed to be in the grounded extension, and the old is guaranteed to be out as a result of the preceding *LocalCounter* move. So, for *LocalLemmaInc(m, l, k)*, the resulting update to *ASPIC*$^+$ is just $\mathcal{K}' = \mathcal{K} \cup \{k\}$, with the corresponding update at the abstract level: $AR'_{LG} = AR_{LG} \cup \{k\}$ and $attacks'_{LG} = attacks_{LG}$.

For *PDefinition*, Proponent's definition, d in support of the reason for treating the counterexample as a monster is added to the AIF. The *ASPIC*$^+$ theory is updated not only with $\mathcal{K}' = \mathcal{K} \cup \{d\}$ but also $\mathcal{R}'_{LG} = \mathcal{R}_{LG} \cup \{d \Rightarrow r\}$. The reason d is guaranteed to be in the grounded extension (again, under a constraint of non-repetition). The abstract argument corresponding to the reason alone is no longer induceable, instead being replaced by the abstract argument representing the complex $[d$ so $r]$. The extant attack between r and the argument α (from m to not-c) is thus replaced by one from δ to α, and likewise that from r to m by one from δ to m. Thus $AR'_{LG} = (AR_{LG} \cup \{d, \delta\}) \setminus \{r\}$ and $attacks'_{LG} = (attacks_{LG} \cup \{(\delta, \alpha), (\delta, m)\}) \setminus \{(r, \alpha), (r, m)\}$ where δ is the abstract argument corresponding to the structure argumentation complex $[d$ so $r]$ and α is that for $[m$ so not-$c]$.

5.2. Critical locutions

Next we move on to moves that reduce the size of the extant theory by introducing conflict into the AIF structure and thence the abstract framework. All are moves of Opponent, as is to be expected. First, *GlobalCounter(m, c)* updates the AIF to introduce not-c in conflict with the conjecture c (as well as introducing m, of course). This conflict introduces attacks between the abstract arguments involving c on the one hand and those involving not-c on the other [16]. The *ASPIC*$^+$ theory is updated such that $\mathcal{K}' = \mathcal{K} \cup \{m, \text{not}{-}c\}$ and $\mathcal{R}'_{LG} = \mathcal{R}_{LG} \cup \{m \Rightarrow \text{not-}c\}$ plus $\overline{c}' = \overline{c} \cup \{\text{not-}c\}$ and $\overline{\text{not-}c}' = \overline{\text{not-}c} \cup \{c\}$. This ensures that c is no longer in the grounded extension. Because the conflict is symmetrical in the AIF, there are bidirectional attacks in the abstract framework so not only is c out of the grounded extension, so too is not-c. This accurately reflects not only that c is knocked out of the current proof, but also that the counterposition is *not* added to the theory. Instead the proof is temporarily suspended – if the dialogue were to stop at this point, there would be no proof: which is exactly the status delivered by the grounded extension of the abstract framework. For *GlobalCounter(m, c)*, we have that $AR'_{LG} = AR_{LG} \cup \{m, \alpha\}$ and $attacks'_{LG} = attacks_{LG} \cup \{(\omega, \alpha), (\alpha, \omega)\}$, where α is the abstract argument corresponding to the structured

argumentation complex [m so not-c] and ω corresponds to the argument [l_i so c] (i.e. that connects all of the lemmas l_i to the conjecture, c).

The *LocalCounter(m, l)* and *HybridCounter(m, l, c)* moves similarly add conflicts, and have similar effects on the abstract framework and thence the grounded extension. For *LocalCounter(m, l)*, $\mathcal{K}' = \mathcal{K} \cup \{m\}$ and $\overline{m}' = \overline{m} \cup \{l\}$ and $\overline{l}' = \overline{l} \cup \{m\}$. At the abstract level, this yields $AR'_{LG} = AR_{LG} \cup \{m\}$ and $attacks'_{LG} = attacks_{LG} \cup \{(l, m), (m, l), (m, \omega)\}$, where ω corresponds to the argument [l_i so c]. The update associated with $HybridCounter(m, l, c)$ is the union of those for *LocalCounter* and *GlobalCounter*, except that the monster, m is itself the counter to the conjecture, *viz.*, $\mathcal{K}' = \mathcal{K} \cup \{m\}$ and $\overline{m}' = \overline{m} \cup \{l, c\}$ and $\overline{l}' = \overline{l} \cup \{m\}$ and $\overline{c}' = \overline{c} \cup \{m\}$. Thus at the abstract level, $AR'_{LG} = AR_{LG} \cup \{m\}$ and $attacks'_{LG} = attacks_{LG} \cup \{(m, l), (l, m), (m, \omega), (\omega, m), (m, c), (c, m)\}$, where ω corresponds to the argument [l_i so c].

For *MonsterReject*, the situation rather mirrors *MonsterBar* in that a reason, s, along with the symmetrical conflict between s and r are added to the AIF. With no resolution between these two, neither s nor r will be in the grounded extension; as a result the abstract argument corresponding to the argument from counterexample to counter-conjecture will be attacked by nothing other than the conjecture, so both will be excluded from the grounded extension. Finally, the new reason, s is adduced as an additional premise in the argument from m to not-c. Thus for $MonsterReject(m, r, d, s, c)$, the $ASPIC^+$ framework is updated so that $\mathcal{K}' = \mathcal{K} \cup \{s\}$ and $\mathcal{R}'_{LG} = \mathcal{R}_{LG} \cup \{s, m \Rightarrow \text{not-}c\}$ plus $\overline{s}' = \overline{s} \cup \{r\}$ and $\overline{r}' = \overline{r} \cup \{s\}$. In the resulting abstract framework, $AR'_{LG} = AR_{LG} \cup \{s, \sigma\}$ and $attacks'_{LG} = attacks_{LG} \cup \{(\delta, s), (s, \delta), (\delta, \sigma), (\sigma, \omega), (\omega, \sigma)\}$, where δ is the abstract argument corresponding to the structured argumentation complex [d so r], σ is [$\{s, m\}$ so not-c] and ω is [l_i so c].

Lastly, *ODefinition*, which is only used as Opponent's substantiation of their *MonsterReject*, introduces a new definition, e and symmetrical conflict between that and d. Again, with no resolution between d and e, neither are in the grounded extension. The update introduced by $ODefinition(m, r, d, s, e)$ on $ASPIC^+$ is thus $\mathcal{K}' = \mathcal{K} \cup \{e\}$ and $\mathcal{R}'_{LG} = \mathcal{R}_{LG} \cup \{e \Rightarrow s\}$ plus $\overline{d}' = \overline{d} \cup \{e\}$ and $\overline{e}' = \overline{e} \cup \{d\}$. In the resulting abstract framework, $AR'_{LG} = (AR_{LG} \cup \{e, \varepsilon\}) \setminus \{s\}$ and $attacks'_{LG} = (attacks_{LG} \cup \{(d, e), (e, d), (\delta, \varepsilon), (\varepsilon, \delta)\}) \setminus \{(s, \delta), (\delta, s)\}$, where δ is the abstract argument corresponding to the structured argumentation complex [d so r] and ε is [e so s].

5.3. Neutral locutions

Two further moves have important updates to the abstract argumentation framework despite not updating the theory structure directly (that is to say, they are important in terms of the dialogical dynamics of LG but have only indirect impact on the theory and its update (indirect inasmuch as constraining the subsequent application of moves that do have direct impact on the theory).

The *Prefer* move, is unique in that it will either expand or contract the grounded extension depending on which player executes it. The protocol demands that $Prefer$ can only be executed after both Proponent and Opponent have offered definitions, which in turn can only occur in a *MonsterBar-PDefinition-MonsterReject-ODefinition* series. First, we consider Opponent's use of *Prefer*, in which Opponent's definition, e, is preferred to Proponent's, d. This ensures that the abstract argument corresponding to e defeats that corresponding to d, and further that the abstract argument ε (corresponding to the structured argumentation complex [e so s]) defeats the abstract argument δ (corresponding to the structured argumentation complex [d so r]). With δ out of the grounded extension, the argument α (the abstract argument corresponding to the argument [m so not-c]) is no longer undercut, thereby ensuring that c is out of the grounded extension. For Proponent's use of the *Prefer* move, the situation is reversed, so that d is preferred to e, and thence that δ is preferred to ε. This in turn ensures that α is out of the grounded extension and with nothing else to attack it, c is in. So after Proponent's $Prefer$ move, $c, r, d \in GE'_{LG}$, whilst not-$c, m, s, e \notin GE'_L G$. The definition of $Prefer(m, r, f, g)$ allows us to capture these abstract consequences independently of the speaker, expressing that f is preferred to g (and in the abstract, that η founded on f is preferred to γ founded on g), regardless of which of f and g are instantiated with d or e. The effect of adding a preference into the structured argumentation is to remove one of the pair of symmetrical attacks in the abstract framework. The $ASPIC^+$ update is thus just $\overline{f}' = \overline{f} \cup \{g\}$ which results at the abstract level, in $AR'_{LG} = AR_{LG}$ and $attacks'_{LG} = attacks_{LG} \setminus \{(g, f), (\gamma, \eta)\}$.

The *ProofDone* move adds a new abstract argument, ω corresponding to the argument from all of the current lemmas, l_i, to the conjecture, c, and uses it to replace the atomic abstract argument corresponding to c. Finally, everything that previously attacked c will now be attacking ω. For $ASPIC^+$, the update is thus simply $\mathcal{R}'_{LG} = \mathcal{R}_{LG} \cup \{l_i \Rightarrow c\}$. The ramifications at the abstract level are more complex: $AR'_{LG} = (AR_{LG} \cup \omega) \setminus \{c\}$ and $attacks'_{LG} = attacks_{LG} \cup (\beta_i, \omega), \forall \beta_i$ s.t. $(\beta_i, c) \in attacks_{LG}$, where ω is the abstract argument corresponding to the structured argumentation complex [l_i so c].

To complete the assessment of semantic update, it is also useful to explain why the remaining four moves have no direct effect at all.

The purpose of *HybridLemmaInc* is to introduce a new conjecture, c', but it is not *HybridLemmaInc* itself that does this, but rather the *Conjecture* move which is required to follow it. Similarly, the abstract argument corresponding to the inference from the new set of lemmas to the new conjecture is handled later by *ProofDone* whilst the exclusion of the lemma that has been incorporated and of the conjecture which has been revised are both handled earlier by the *HybridCounter* or *GlobalLemmaInc* which must have preceded it. The update associated with *HybridLemmaInc* itself, therefore, is null (i.e. $AR'_{LG} = AR_{LG}$ and $attacks'_{LG} = attacks_{LG}$).

The *MonsterAccept* move allows Opponent to reiterate what has already been said, and so does not update the semantics any more than it updates the underlying AIF structure.

Table 1
Summary of the syntactic and semantic updates in *LG*.

Move	AIF update	$ASPIC^+$ update			AF_{LG} update	
		$\mathcal{K}$	$\mathcal{R}_{LG}$	$^-$	AR_{LG}	$attacks_{LG}$
Conjecture(c)	$+\ c$	$\cup\{c\}$	$\emptyset$	$\emptyset$	$\cup\{c\}$	$\emptyset$
Lemma(l)	$+\ l$	$\cup\{l\}$	$\emptyset$	$\emptyset$	$\cup\{l\}$	$\emptyset$
ProofDone	$+$ RA(l_i, c)	$\emptyset$	$\cup\{l \Rightarrow c\}$	$\emptyset$	$\cup\{\omega\} \setminus \{c\}$	$\cup(\beta_i, \omega)$, $\forall\beta_i$ s.t. $(\beta_i, c) \in attacks_{LG}$
LocalCounter(m, l)	$+\ m$, CA(m, l), CA(l, m)	$\cup\{m\}$	$\emptyset$	$\cup\{(m, l), (l, m)\}$	$\cup\{m\}$	$\cup\{(l, m), (m, l), (m, \omega)\}$
HybridCounter(m, l, c)	$+\ m$, CA(m, l), CA(l, m), CA(m, c), CA(c, m)	$\cup\{m\}$	$\emptyset$	$\cup\{(l, m), (m, l), (c, m), (c, m)\}$	$\cup\{m\}$	$\cup\{(m, l), (l, m), (m, \omega), (\omega, m), (m, c), (c, m)\}$
GlobalCounter(m, c)	$+\ m$, not-c, RA$(m,$ not-$c)$, CA$(c,$ not-$c)$, CA(not-c, c)	$\cup\{m,$ not-$c\}$	$\cup\{m \Rightarrow$ not-$c\}$	$\cup\{(c,$ not-$c),$ (not-$c, c)\}$	$\cup\{m, \alpha\}$	$\cup\{(\omega, \alpha), (\alpha, \omega)\}$
PiecemealExclusion(b)	$+\ b$	$\cup\{b\}$	$\emptyset$	$\emptyset$	$\cup\{b\}$	$\emptyset$
StrategicWithdrawal(r, c)	$+\ r$, CA(r, c), CA(c, r)	$\cup\{r\}$	$\emptyset$	$\cup\{(r, c), (c, r)\}$	$\cup\{r\}$	$\cup\{(r, \omega), (\omega, r)\}$
LocalLemmaInc(m, l, k)	$+\ k$, MA(l, k)	$\cup\{k\}$	$\emptyset$	$\emptyset$	$\cup\{k\}$	$\emptyset$
HybridLemmaInc(m, l)	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
GlobalLemmaInc(m, k)	$+\ k$, CA(m, k), CA(k, m)	$\cup\{k\}$	$\emptyset$	$\emptyset$	$\cup\{k\}$	$\cup\{(k, \alpha), (m, k), (k, m)\}$
MonsterBar(m, c, r)	$+\ r$, CA$(r,$ RA$(m,$ not-$c))$	$\cup\{r\}$	$\emptyset$	$\cup\{(m \Rightarrow$ not-$c, r)\}$	$\cup\{r\}$	$\cup\{(r, \alpha)\}$
MonsterAdjust(m, r)	$+\ r$, CA(r, m), CA(m, r)	$\cup\{r\}$	$\emptyset$	$\cup\{(m, r)\} \setminus \{(r, m)\}$	$\cup\{r\}$	$\cup\{(r, m)(r, \alpha)\}$
PDefinition(m, r, d)	$+\ d$, RA(d, m)	$\cup\{d\}$	$\cup\{d \Rightarrow r\}$	$\emptyset$	$\cup\{d, \delta\} \setminus \{r\}$	$\cup\{(\delta, m), (\delta, \alpha)\} \setminus \{(r, m), (r, \alpha)\}$
ODefinition(m, r, d, s, e)	$+\ e$, RA(s, e), CA(e, d), CA(d, e)	$\cup\{e\}$	$\cup\{e \Rightarrow s\}$	$\cup\{(e, d), (d, e)\}$	$\cup\{e, \varepsilon\} \setminus \{s\}$	$\cup\ \{(d, e), (e, d), (\delta, \varepsilon), (\varepsilon, \delta)\} \setminus \{(s, \delta), (\delta, s)\}$
Prefer(m, r, f, g)	$+$ PA(f, g)	$\emptyset$	$\emptyset$	$\setminus\{(f, g)\}$	$\emptyset$	$\setminus\{(g, f), (\gamma, \eta)\}$
MonsterAccept(m, r)	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
MonsterReject(m, r, d, s, c)	$+\ s$, RA$(\{s, m\},$ not-$c)$, CA(s, r), CA(r, s)	$\cup\{s\}$	$\cup\{\{s, m\} \Rightarrow$ not-$c\}$	$\cup\{(s, r), (r, s)\}$	$\cup\{s, \sigma\}$	$\cup\{(\delta, s), (s, \delta), (\sigma, \omega), (\omega, \sigma), (\delta, \sigma)\}$
Accept	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
Surrender	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

In addition there are the two terminating moves, *Accept* and *Surrender* neither of which effect material update on the AIF structure. Instead, they mark the conclusion of the dialogical process.

In short, *HybridLemmaInc*, *MonsterAccept*, *Accept* and *Surrender* all play a role in the dialogical structure, but their place in the dialogical exchange ensures that any appropriate updates with resulting semantic changes are executed by other moves in the dialogical sequence.

The complete set of updates, both syntactic and resultant semantic, for all moves in *LG* is summarised in Table 1, which also shows, in the final column, the direct impact on the resulting grounded extension – though this omits potential reinstatements and other indirect effects of the updates described in columns three and four.

6. Implementation: from the theoretical model to computational model

We have shown how argumentation semantics can, in theory, furnish us with the most up-to-date status of a theory in a Lakatos Game. All the components of this pipeline [88] are also implemented – so, execution of Lakatos Games requires no new development other than the creation of a Dialogue Game Description Language (DGDL) [105] specification to capture the rules described in Section 3 and the updates described in Section 4.

A DGDL specification consists of three main parts; composition, rules and interactions. The composition describes the general features of the dialogue game, and, in the case of LG, the composition is as shown in Listing 1.

Firstly, we specify that a turn can consist of multiple moves and that the ordering is liberal, i.e. that there are not single alternating moves made by each participant. The roles of Participant and Opponent are then given with a limitation that only these two players are possible. Finally, we specify the commitment stores that will be used to store the conjecture and the lemmas that will comprise the proof.

The LG specification contains only one rule, a starting rule that is triggered when the dialogue begins and specifies that the dialogue starts with Proponent making a ‘Conjecture’ move.

```
turns{magnitude:multiple, ordering:liberal}
roles{Proponent, Opponent}
players{min:2, max:2}
player{id:Proponent}
player{id:Opponent}
store{id:Conjecture, owner:Proponent,
      structure:set, visibility:public, {""}}
store{id:Lemmas, owner:Proponent,
      structure:set, visibility:public, {""}}
```

Listing 1: LG DGDL composition.

```
interaction{Conjecture, {c}, Asserting, {c}, "$c is the conjecture",
    {
    if { size(Lemmas, Proponent, !empty) } then
        { move(add, next, ProofDone, Proponent)
        & move(add, next, Lemma, {l}, Proponent)
        & store(empty, Conjecture, Proponent)
        & store(add, {c}, Conjecture, Proponent)
        }
    else
        { move(add, next, Lemma, {l}, Proponent)
        & store(empty, Conjecture, Proponent)
        & store(add, {c}, Conjecture, Proponent)
        }
    }
}
```

Listing 2: The Conjecture interaction.

```
interaction{GlobalCounter, {m,c},
            Asserting, {m}, Asserting, {!c},
            Contradicting, {<{c},{!c}>, DefaultConflict},
            Contradicting, {<{!c},{c}>, DefaultConflict},
            Arguing, {<{m},{!c}>, DefaultSupport}, "$m is a counter to $c",
        { move(add, next, PiecemealExclusion, {b}, Proponent)
        & move(add, next, StrategicWithdrawal, {r, c}, Proponent)
        & move(add, next, MonsterBar, {m,c,r}, Proponent)
        & move(add, next, MonsterAdjust, {m,r}, Proponent)
        & move(add, next, Surrender, Proponent)
        & move(add, next, GlobalLemmaInc,{m,l}, Proponent)
        }
    }
```

Listing 3: The GlobalCounter interaction.

The remainder of the LG DGDL specification then consists of the interactions available and the effects which they have on the dialogue. A simple example of an interaction description is Proponent's first move, 'Conjecture' (Listing 2).

In this example, the name of the move is given first, followed by its content and the fact that Proponent is asserting the conjecture. The string "$c is the conjecture" is a description of what the move is doing, used to allow users to identify it. The body of the interaction looks at whether the Lemmas commitment store is empty, and if not, allows Proponent to perform a ProofDone move next. Regardless of the state of the Lemmas store, Proponent also has the option to perform the Lemma move at this point. Finally, in both cases, any existing conjecture is removed from the Conjecture store and the new conjecture is added.

Another example of a more complex interaction, 'GlobalCounter', is given in Listing 3.

Here Opponent is giving a counterexample, 'm', to the conjecture, 'c'. This counterexample is arguing in support of the negation of the conjecture, '!c'.

Having specified the LG system in DGDL, this specification can then be processed by the Dialogue Game Execution Platform (DGEP) [14]. DGEP allows participants to take part in dialogues following the rules specified by a DGDL protocol. When a new dialogue is initiated, DGEP allows users to join in the roles available, and any initial rules are processed. From this point on, DGEP maintains the legal move list for each participant, based on the rules and interactions: a rule is executed when it is in scope, and an interaction when it is moved by a player during the game.

DGEP provides a range of web service interfaces, allowing a user to both perform interactions and get information about the current dialogue state (for example, their list of available moves). These web services can then be used by either software agents playing the roles of specific participants, or by graphical interfaces allowing human users to take part in the discussion. Arvina [56] is one such graphical interface, and a screenshot of a LG protocol dialogue taking place in Arvina can be seen in Fig. 11.

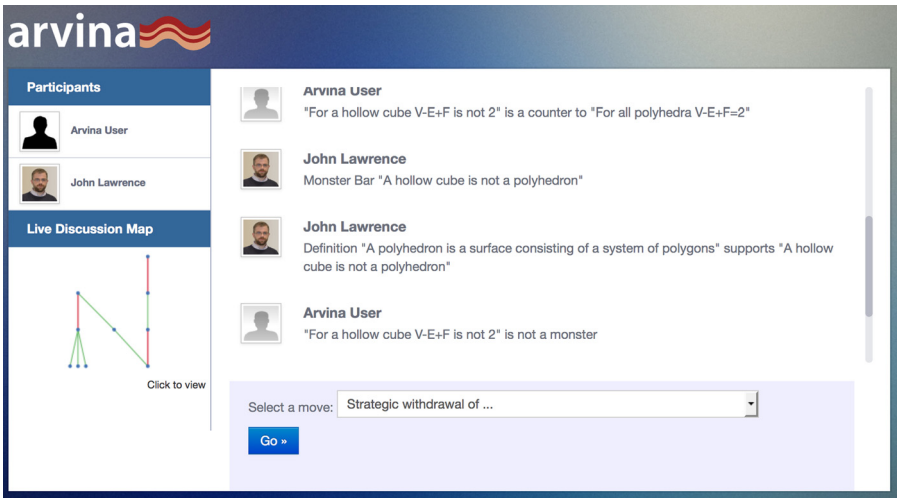


Fig. 11. Arvina screenshot.

One of the advantages of using DGEP is that it not only provides a robust platform for the execution of a dialogue protocol, but also creates argument structures as a side effect. AIF infrastructure for querying and updating is available in the webservices of AIFdb[16] [57]; conversion to *ASPIC*$^+$ and induction of abstract frameworks is handled by TOAST [90]; and calculation of acceptability semantics is performed by Dung-O-Matic [31]. (We use these tools for convenience because they work directly with AIF and *ASPIC*$^+$ data structures, though many other systems perform comparable computation using various alternative approach – Tweety [93] using defeasible logic programming, ASPARTIX [33] using answer set programming and ArgSemSAT [24] using SAT solving.)

Arvina has been demonstrated to provide a practical stepping stone towards mixed-initiative argumentation [89], a type of collaborative intelligence [35] or human–agent collective [49]. Where agents have knowledge bases populated by pre-existing arguments, they can contribute to an evolving Lakatos game on a level playing field with human participants. Indeed, DGEP makes no distinction between types of clients, whether human or artificial. The Arvina system, the Lakatos game, and agents containing several small suitable knowledge bases are available online at http://arvina.arg-tech.org.

Finally, the AIF structures created as a side-effect of executing the DGDL specification for LG using Arvina can be submitted to the TOAST system [90] which implements (i) the translation from AIF to *ASPIC*$^+$; (ii) the induction of abstract frameworks according to the *ASPIC*$^+$ definitions; and (iii) the computation of semantics over those structures, including grounded, by calling DungOMatic [31]. Sample results from TOAST running over structures created by Arvina and DGEP executing the Lakatos Game DGDL specification are shown in Fig. 12.

7. Execution: collective proof as argumentation

In this section we look at a fully worked example dialogue, considering strategies of global counter and monster-barring (see Section 7.2), and then local counter and local lemma incorporation (see Section 7.3) in greater depth. A proof idea for Euler's conjecture is presented by the Teacher on the second page *Proofs and Refutations* [55]. The proof outline comes from [23], and consists of three steps (for a diagrammatic representation of these steps, carried out on the cube, see Fig. 13, taken from [55, p. 8]). Note that in (1-i), GAMMA's reference to "my counterexample" is made on behalf of both ALPHA and GAMMA who speak in the role of Opponent throughout, whereas TEACHER and DELTA speak in the role of Proponent.

(1) a. TEACHER: We arrived at a conjecture concerning polyhedra, namely that for all polyhedra, $V - E + F = 2$, where V is the number of vertices, E the number of edges and F the number of faces. ... ... I have one [a proof]. It consists of the following thought experiment.
b. TEACHER: Step 1 [described above, as lemmas].
c. TEACHER: Step 2.
d. TEACHER: Step 3.
e. TEACHER: Thus we have proved our conjecture.
f. ALPHA: I have a counterexample … Imagine a solid bounded by a pair of nested cubes – a pair of cubes, one of which is inside, but does not touch the other. … for each cube $V - E + F = 2$, so that for the hollow cube $V - E + F = 4$.
g. DELTA: This pair of nested cubes is not a polyhedron at all. … It is a *monster*, … not a counterexample.

[16] http://www.aifdb.org.

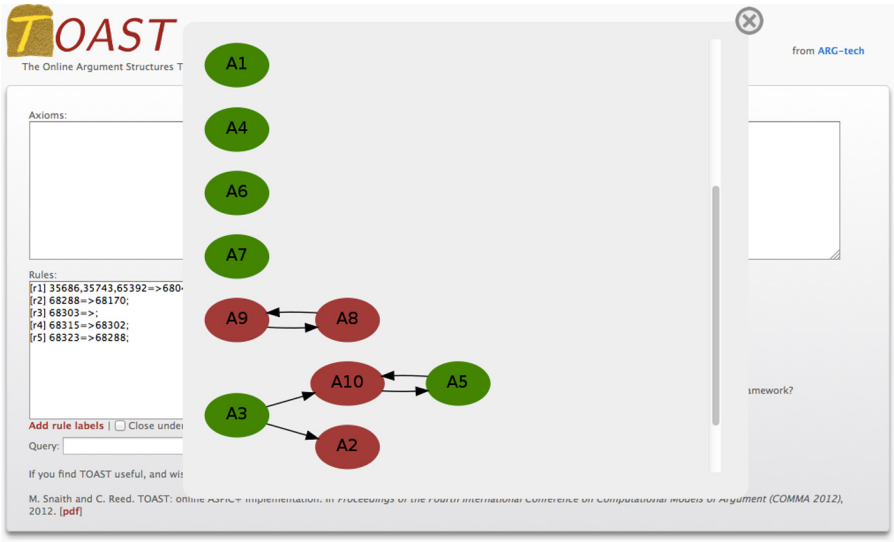


Fig. 12. TOAST running on a Lakatos dialogue example.

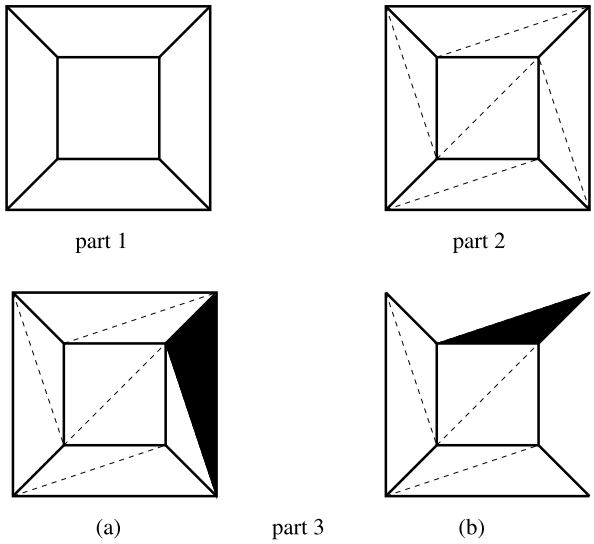


Fig. 13. Given the cube, after removing a face and stretching it flat, we are left with the network in part 1. After triangulating, we get part 2. When removing a triangle, we either remove one edge and one face, or two edges, one vertex and a face – shown in parts 3(a) and (b) respectively.

h. DELTA: A polyhedron is a surface consisting of a system of polygons.
i. GAMMA: My counterexample is a solid bounded by polygonal faces.
j. GAMMA: A polyhedron is a solid whose surface consists of polygonal faces.
k. TEACHER: …For the moment let us accept Delta's definition. Can you refute our conjecture now if by polyhedron we mean a surface?
l. GAMMA: I propose a trivial counterexample. Take the triangular network which results from performing the first two operations on a cube. Now if I remove a triangle from the *inside* of this network, as one might take a piece of the jigsaw puzzle, I remove one triangle without removing a single edge or vertex. So the third lemma is false – and not only in the case of the cube, but for *all* polyhedra except the tetrahedron, in the flat network of which all the triangles are boundary triangles. …
m. TEACHER: I no longer contend that *the removal of any triangle follows one of the two patterns mentioned*, but merely that *at each stage of the removing operation the removal of any boundary triangle follows one of these patterns.* … All that I have to do is to insert a single word in my third step, to wit, that 'from the triangulated network we now remove the *boundary* triangles one by one'.
n. TEACHER: … I can easily (…) *improve the proof*, by replacing the false lemma by a slightly modified one, which your counterexample will not refute.

Table 2
Proponent's construction of the initial proof.

Formalisation		Representation		Evaluation	
LG system	*Dialogue theory*	*AIF*	*ASPIC*$^+$	AF_{LG}	GE_{LG}
Proponent – Proposal of initial proof					
(1-a) *Conjecture*(C) **[L1.1, S1]**	C	$\mathbf{I}_1$: For all polyhedra, $V-E+F=2$ [C]	$\mathcal{K}=\{C\}$ $\mathcal{R}_{LG}=\{\}$ $^-=\{\}$	$AR_{LG}=\{C\}$ $att_{LG}=\{\}$	C
(1-b) *Lemma*(L_1) **[L1.2, S2.2]**	C L_1	$\mathbf{I}_2$: Any polyhedron, after having a face removed, can be stretched flat on the blackboard [L_1]	$\mathcal{K}=\{C, L_1\}$ $\mathcal{R}_{LG}=\{\}$ $^-=\{\}$	$AR_{LG}=\{C, L_1\}$ $att_{LG}=\{\}$	C L_1
(1-c) *Lemma*(L_2) **[L1.2, S2.1b]**	C L_1 L_2	$\mathbf{I}_3$: In triangulating a map, one will always get a new face for any new edge [L_2]	$\mathcal{K}=\{C, L_1, L_2\}$ $\mathcal{R}_{LG}=\{\}$ $^-=\{\}$	$AR_{LG}=\{C, L_1, L_2\}$ $att_{LG}=\{\}$	C L_1 L_2
(1-d) *Lemma*(L_3) **[L1.2, S2.1b]**	C L_1 L_2 L_3	$\mathbf{I}_4$: if we drop the triangles one by one from a triangulated map, there are only two alternatives – the disappearance of one edge or else of two edges and a vertex [L_3]	$\mathcal{K}=\{C, L_1, L_2, L_3\}$ $\mathcal{R}_{LG}=\{\}$ $^-=\{\}$	$AR_{LG}=\{C, L_1, L_2, L_3\}$ $att_{LG}=\{\}$	C L_1 L_2 L_3
(1-e) *ProofDone*() **[L1.3, S2.1a]**	L_1 L_2 L_3 L_1, L_2, L_3 so C	$\mathbf{RA}_1$: ($\{\mathrm{I}_2, \mathrm{I}_3, \mathrm{I}_4\}$, I_1) [($\{L_1, L_2, L_3\}$, C)]	$\mathcal{K}=\{C, L_1, L_2, L_3\}$ $\mathcal{R}_{LG}=\{(L_1, L_2, L_3)\Rightarrow C\}$ $^-=\{\}$	$AR_{LG}=\{L_1, L_2, L_3, \omega\}$ $att_{LG}=\{\}$	L_1 L_2 L_3 ω

In the remainder of this section, we work step by step through this dialogue and show how our protocol both, covers all of the moves made, and, at the same time, produces updates to the generated the Argument Interchange Format structure and argumentation structure which capture the status of the proof after each turn of the dialogue.

7.1. Introducing the proof

The first five moves in the above dialogue transcript, (1-a) to (1-e), correspond to the introduction of the initial proof by Proponent. Table 2 shows how the execution of these steps using the LG dialogue system (column 1 of the table) updates a shared theory, understood by the participants as the current state of the proof (according to the definitions in Section 3.2) (in column 2). The AIF structures added by these moves are shown in column 3.[17] The current *ASPIC*$^+$ framework caused by mapping the complete new AIF structures according to [16] are shown in column 4. The abstract argumentation framework caused by inducing from the new *ASPIC*$^+$ framework a new AF are shown in column 5, and finally the grounded extension computed over that AF is shown in the last column.

In (1-a), Proponent presents their conjecture: "For all polyhedra, $V-E+F=2$", this corresponds to the *Conjecture(C)* move in the LG protocol governed by the locution rule **L1.1** and structural rule **S1**. The current state of proof consists now of the conjecture C which adds I_1, a first node, to the AIF structure depicted on the left hand side in Fig. 14. This also adds C as an argument to the *ASPIC*$^+$ framework, and to the abstract argumentation framework. As there are no conflicts at this stage, C is in the grounded extension.

The next three moves (1-b), (1-c), and (1-d), present the three lemmas that the Proponent is using to support their conjecture (the full text of these lemmas can be seen in Table 2). This introduces three lemmas L_1, L_2, L_3 into dialogue theory, and three new nodes I_2, I_3, I_4 into AIF structure and argumentation framework in Fig. 14.

Finally, in (1-e), the Proponent announces that they have completed constructing their proof, making the *ProofDone* move. Performing *ProofDone* not only shows that the proof is complete allowing the Opponent to raise any challenges, but also creates a new rule of inference (RA_1) whereby representing that the given lemmas support the conjecture. This creates a new abstract argument, ω, corresponding to the proof structure, and, as there are no attacks at this stage, ω and the three given lemmas are all in the grounded extension (Fig. 14). The updates performed during these moves are shown in Table 2.

[17] As the AIF updates only add to the structure, we only show those components added for each move in the table. For all other columns, the cumulative state is shown in each row.

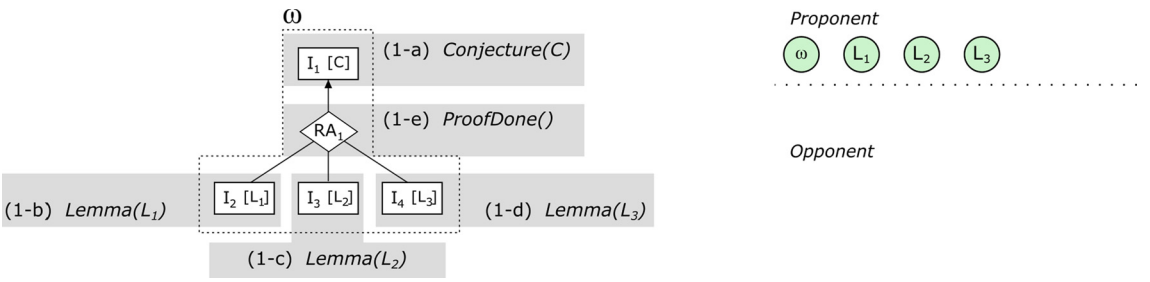


Fig. 14. The update made to the Argument Interchange Format structure (on the left-hand side) and the argumentation framework (on the right-hand side) upon Proponent's introduction of the initial proof. On the left hand side, nodes labelled as I_1–I_4 are information nodes which correspond to the statements exchanged during the first five moves of the dialogue (1-a)–(1e) (the symbols from LG system, C, L_1–L_3, are given in the brackets). The moves specified according to the LG systems are marked by grey area. The nodes RA_m represent inferences and CA_n (on the next figures) – conflicts. The right hand side of figure shows the abstract framework created as a result of the update to the AIF structure (nodes at the top are Proponent's arguments and nodes on the bottom – Opponent's). The nodes in bold line denote arguments which are acceptable (in GE), while the nodes in dotted line denote arguments which are unacceptable (out of GE) at a given stage of the dialogue. Finally, some moves create new arguments to be added or replace other arguments in the framework: in this case we mark the fragment of the AIF structure and label it with new abbreviation such as ω in this figure.

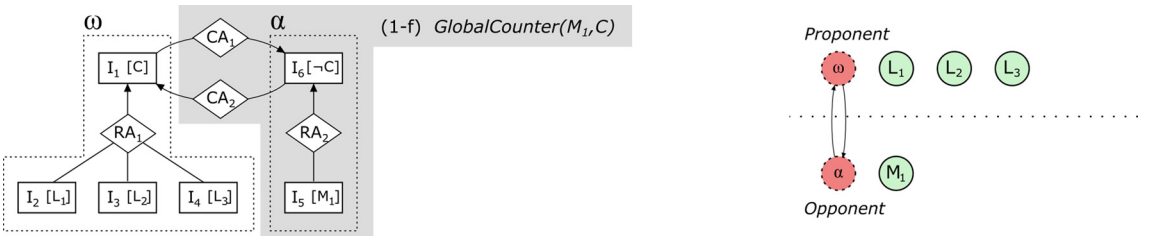


Fig. 15. The update made to the Argument Interchange Format structure (on the left-hand side) and the argumentation framework (on the right-hand side) upon Opponent's counter to the conjecture (1-f).

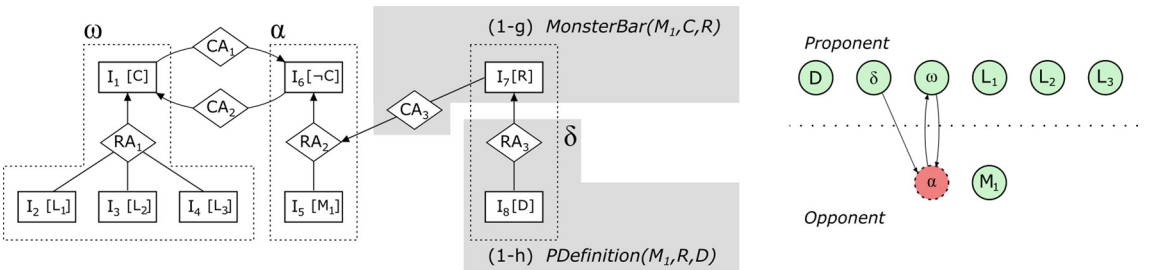


Fig. 16. The update made to the Argument Interchange Format structure and the argumentation framework upon Proponent's rejection of M_1 as a counter (1-g), and definition supporting this (1-h).

7.2. First strategy of testing the proof: countering the conjecture

Once the Proponent has presented the initial proof, the Opponent makes their first attack. Here, the Opponent provides a counter to the conjecture, which the Proponent rejects as not fitting the definition of a polyhedron. The updates carried out in this section of the dialogue are summarised in Table 3.

The counter begins in (1-f), following the rules **L2.3** and **S4.1**, the Opponent uses *GlobalCounter* to introduce the counterexample of a hollow cube (i.e. M_1 or I_5): "For the hollow cube $V - E + F = 4$", to the conjecture "For all polyhedra, $V - E + F = 2$" (thus new conflict nodes CA_1 and CA_2 in the updated AIF structure in Fig. 15). More specifically, they use this counterexample as a premise to justify (RA_2) the counter-conjecture (not-C, I_6): "It is not the case that for all polyhedra, $V - E + F = 2$". This results in the creation of a new abstract argument in Fig. 15, α, which is in mutual conflict with the abstract argument, ω, representing the proposed proof, and hence ω is no longer in the grounded extension.

The Proponent defends this challenge in (1-g), using *MonsterBar*, to show that this is not a valid counterexample, claiming that, whilst the hollow cube does indeed have the property "$V - E + F = 4$", it is not a polyhedron (R or I_7). Following rule **S10**, the Proponent must then support this statement (RA_3), using the *PDefinition* move. As such, the Proponent provides a definition of a polyhedron: "A polyhedron is a surface consisting of a system of polygons", which excludes the hollow cube (D or I_8). The resulting abstract argument, δ, conflicts with α, returning ω, to the grounded extension (see Fig. 16).

At this stage in the dialogue, Opponent could choose to agree with Proponent that this is not a valid counterexample (rule **S12.1**). However, in this example Opponent decides to continue the attack (rule **S12.2**), first making the move *MonsterReject*,

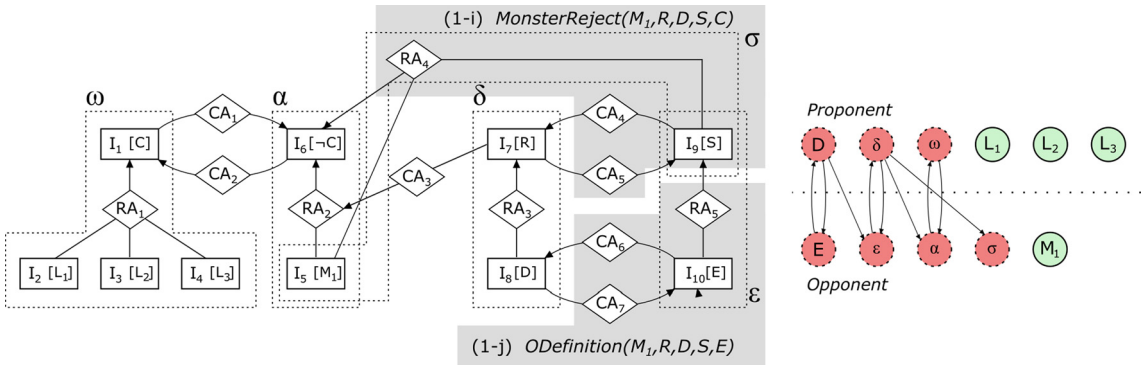


Fig. 17. The update made to the Argument Interchange Format structure and the argumentation framework upon Opponent's rejection of the Proponents definition (1-i) and proposal of an alternative definition (1-j).

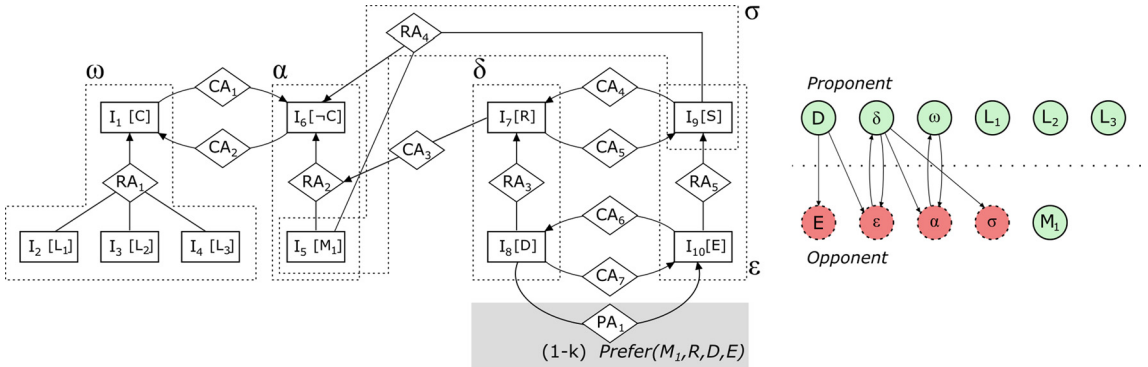


Fig. 18. The update made to the Argument Interchange Format structure and the argumentation framework upon Proponent's definition being preferred (1-k).

to reject Proponent's defence by using I_9 (S), and CA_4 and CA_5, and then providing an alternative definition of a polyhedron (I_10 or E), which supports I_9, using *ODefinition*. As we now have two conflicting definitions, the Proponents reason is no-longer sufficient to mean that the counter is rejected, and, as such, ω is once again out of the grounded extension (Fig. 17).

Progress in the dialogue reaches an impasse until one of the party excepts the definition proposed by the other. In the case of our example, the Proponent's definition is preferred, and the Opponent makes the *Prefer* move to indicate this. In the AIF structure, this results in a preference (PA_1 node) showing that D is preferred to E, and, as such, the conflict from E to D in the abstract argument structure is removed. This impacts the grounded extension, by re-instating all of the Proponents arguments, meaning that the proponent has successfully defended against the counter (Fig. 18).

7.3. Second strategy of testing the proof: countering a lemma

With Proponent having successfully defended against the first attack of the proof, Opponent now makes a further attack. This time Opponent raises a counterexample to one of the lemmas. Proponent accepts that this counter is valid and replaces the attacked lemma with a revised version incorporating the counterexample. The updates carried out in this section of the dialogue are summarised in Table 4. (As this section follows on from the previous counter, the definitions introduced there remain a part of the structure, but are omitted here as they have no impact on the remainder of the dialogue.)

The second counter begins in (1-l), following the rules **L2.1** and **S16.3b** (as this counter follows directly from the previous *Prefer* move), the Opponent uses *LocalCounter* to introduce the counterexample of a cube, explaining that, if a triangle is removed from inside the triangular network produced by performing the first two steps on this polyhedron, then no edges on vertices are removed (I_11 in the updated AIF structure in Fig. 19). This is in conflict (CA_8 and CA_9) with the lemma, L_3, one of the premises of the abstract argument, ω, representing the proposed proof, and hence ω is no longer in the grounded extension (Fig. 19).

In (1-m) the Proponent accepts the validity of the counterexample raised by the Opponent, and revises the attacked lemma, L_3, via *LocalLemmaInc* to take this into account (a node MA_1 represents the rephrase relation between the new lemma and the old lemma). This new lemma, K or I_12 in the AIF structure, adapts L_3 by insisting that triangles may only be removed from the boundary of the network. At this stage in the defence, ω is still out of the grounded extension, and the only change is the addition of K (Fig. 20).

Table 3
First strategy of testing the proof: countering the conjecture.

Formalisation		Representation		Evaluation	
LG system	*Dialogue theory*	*AIF*	*ASPIC*$^+$	AF_{LG}	GE_{LG}
Opponent – Challenge 1					
(1-f) *GlobalCounter* (M_1, C) **[L2.3, S4.1]**	L_1 L_2 L_3 M_1	$\mathbf{I}_5$: For the hollow cube $V - E + F = 4$ $[M_1]$ $\mathbf{I}_6$: It is not the case that for all polyhedra, $V - E + F = 2$ [not-C] $\mathbf{CA}_1$: (I_1, I_6) [(C, not-C)] $\mathbf{CA}_2$: (I_6, I_1) [(not-C, C)] $\mathbf{RA}_2$: (I_5, I_6) [(M_1, not-C)]	$\mathcal{K} = \{C, L_1, L_2, L_3, M_1, \text{not-}C\}$ $\mathcal{R}_{LG} = \{(L_1, L_2, L_3) \Rightarrow C, M_1 \Rightarrow \text{not-}C\}$ $^- = \{(C, \text{not-}C), (\text{not-}C, C)\}$	$AR_{LG} = \{L_1, L_2, L_3, \omega, M_1\}$ $att_{LG} = \{(\omega, \alpha), (\alpha, \omega)\}$	L_1 L_2 L_3 M_1
Proponent – Defence 1					
(1-g) *MonsterBar* (M_1, C, R) **[L5.1, S5.3]**	L_1 L_2 L_3 M_1 R L_1, L_2, L_3 so C	$\mathbf{I}_7$: This pair of nested cubes is not a polyhedron $[R]$ $\mathbf{CA}_3$: (I_7, RA_2) [(R, RA_2)]	$\mathcal{K} = \{C, L_1, L_2, L_3, M_1, \text{not-}C, R\}$ $\mathcal{R}_{LG} = \{(L_1, L_2, L_3) \Rightarrow C, M_1 \Rightarrow \text{not-}C\}$ $^- = \{(C, \text{not-}C), (\text{not-}C, C), (M_1 \Rightarrow \text{not-}C, R)\}$	$AR_{LG} = \{L_1, L_2, L_3, \omega, M_1, \alpha, R\}$ $att_{LG} = \{(\omega, \alpha), (\alpha, \omega), (R, \alpha)\}$	L_1 L_2 L_3 M_1 R ω
(1-h) *PDefinition* (M_1, R, D) **[L5.3, S10]**	L_1 L_2 L_3 M_1 D so R D L_1, L_2, L_3 so C	$\mathbf{I}_8$: A polyhedron is a surface consisting of a system of polygons $[D]$ $\mathbf{RA}_3$: (I_8, I_7) [(D, R)]	$\mathcal{K} = \{C, L_1, L_2, L_3, M_1, \text{not-}C, R, D\}$ $\mathcal{R}_{LG} = \{(L_1, L_2, L_3) \Rightarrow C, M_1 \Rightarrow \text{not-}C, D \Rightarrow R\}$ $^- = \{(C, \text{not-}C), (\text{not-}C, C), (M_1 \Rightarrow \text{not-}C, R)\}$	$AR_{LG} = \{L_1, L_2, L_3, \omega, M_1, \alpha, \delta, D\}$ $att_{LG} = \{(\omega, \alpha), (\alpha, \omega), (\delta, \alpha)\}$	L_1 L_2 L_3 M_1 δ D ω
Opponent – Challenge 2					
(1-i) *MonsterReject* (M_1, R, D, S, C) **[L6.2, S12.2]**	L_1 L_2 L_3 M_1 D	$\mathbf{I}_9$: My counterexample is a solid bounded by polygonal faces $[S]$ $\mathbf{CA}_4$: (I_9, I_7) [(S, R)] $\mathbf{CA}_5$: (I_7, I_9) [(R, S)] $\mathbf{RA}_4$: ({I_5, I_9}, I_6) [({M_1, S}, not-C)]	$\mathcal{K} = \{C, L_1, L_2, L_3, M_1, \text{not-}C, R, D, S\}$ $\mathcal{R}_{LG} = \{(L_1, L_2, L_3) \Rightarrow C, M_1 \Rightarrow \text{not-}C, D \Rightarrow R, (M_1, S) \Rightarrow \text{not-}C\}$ $^- = \{(C, \text{not-}C), (\text{not-}C, C), (M_1 \Rightarrow \text{not-}C, R), (S, R), (R, S)\}$	$AR_{LG} = \{L_1, L_2, L_3, \omega, M_1, \alpha, \delta, D, S, \sigma\}$ $att_{LG} = \{(\omega, \alpha), (\alpha, \omega), (\delta, \alpha), (\delta, S), (S, \delta), (\delta, \sigma)\}$	L_1 L_2 L_3 M_1 D
(1-j) *ODefinition* (M_1, R, D, S, E) **[L5.4, S14]**	L_1 L_2 L_3 M_1	$\mathbf{I}_{10}$: A polyhedron is a solid whose surface consists of polygonal faces $[E]$ $\mathbf{CA}_6$: (I_{10}, I_8) [(E, D)] $\mathbf{CA}_7$: (I_8, I_{10}) [(D, E)] $\mathbf{RA}_5$: (I_{10}, I_9) [(E, S)]	$\mathcal{K} = \{C, L_1, L_2, L_3, M_1, \text{not-}C, R, D, S, E\}$ $\mathcal{R}_{LG} = \{(L_1, L_2, L_3) \Rightarrow C, M_1 \Rightarrow \text{not-}C, D \Rightarrow R, (M_1, S) \Rightarrow \text{not-}C, E \Rightarrow S\}$ $^- = \{(C, \text{not-}C), (\text{not-}C, C), (M_1 \Rightarrow \text{not-}C, R), (S, R), (R, S), (E, D), (D, E)\}$	$AR_{LG} = \{L_1, L_2, L_3, \omega, M_1, \alpha, \delta, D, \sigma, E, \varepsilon\}$ $att_{LG} = \{(\omega, \alpha), (\alpha, \omega), (\delta, \alpha), (D, E), (E, D), (\delta, \varepsilon), (\varepsilon, \delta), (\delta, \sigma)\}$	L_1 L_2 L_3 M_1
Opponent – Concede definition					
(1-k) *Prefer* (M_1, R, D, E) **[L2.3, S4.1]**	L_1 L_2 L_3 M_1 D so R L_1, L_2, L_3 so C	$\mathbf{PA}_1$: (I_8, I_{10}) [(D, E)]	$\mathcal{K} = \{C, L_1, L_2, L_3, M_1, \text{not-}C, R, D, S, E\}$ $\mathcal{R}_{LG} = \{(L_1, L_2, L_3) \Rightarrow C, M_1 \Rightarrow \text{not-}C, D \Rightarrow R, (M_1, S) \Rightarrow \text{not-}C, E \Rightarrow S\}$ $^- = \{(C, \text{not-}C), (\text{not-}C, C), (M_1 \Rightarrow \text{not-}C, R), (S, R), (R, S), (E, D)\}$	$AR_{LG} = \{L_1, L_2, L_3, \omega, M_1, \alpha, \delta, D, \sigma, E, \varepsilon\}$ $att_{LG} = \{(\omega, \alpha), (\alpha, \omega), (\delta, \alpha), (D, E), (\delta, \varepsilon), (\varepsilon, \delta), (\delta, \sigma)\}$	L_1 L_2 L_3 M_1 D δ ω

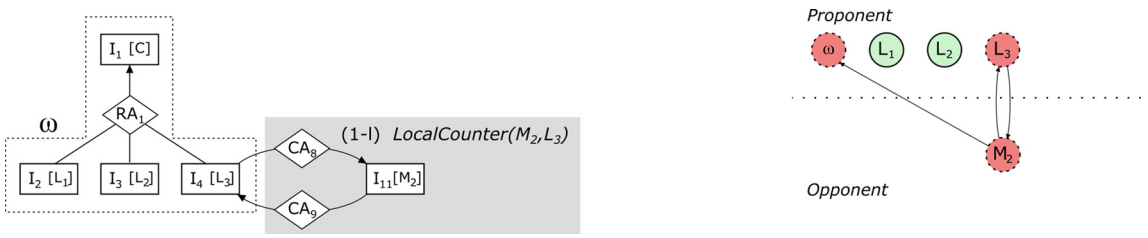


Fig. 19. The update made to the Argument Interchange Format structure (on the left-hand side) and the argumentation framework (on the right-hand side) upon Opponent's counter of L_3 (1-l).

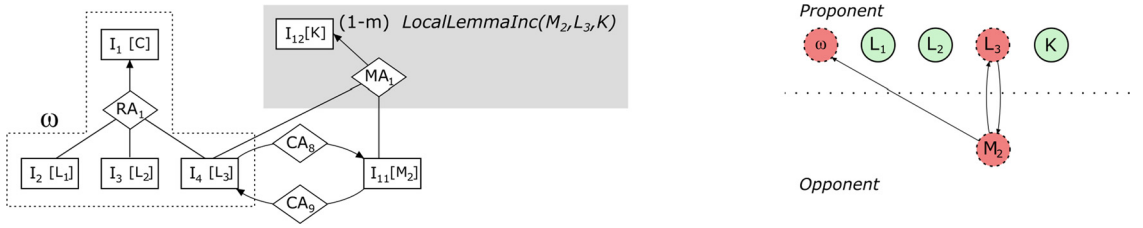


Fig. 20. The update made to the Argument Interchange Format structure and the argumentation framework upon Proponent's introduction of a new lemma, K, replacing L_3 (1-m).

Fig. 21. The update made to the Argument Interchange Format structure and the argumentation framework upon Proponent's creation of the revised proof (1-n).

Having introduced the revised lemma, K, the Proponent now performs the *ProofDone* move to incorporate this new lemma into the proof structure. The result of this move is to create a new rule of inference, RA_6, between the current lemmas (L_1, L_2, K) and the conjecture (C), this produces a new abstract argument ω', which replaces ω as the current proof (Fig. 21). The final grounded extension along with the text of each argument, can be seen in Table 5.

8. Conclusions

In this paper we propose a way to narrow the gap between human and machine proof-construction, in order to promote mainstream acceptance and use of automated theorem provers by mathematicians. Philosophical, sociological and educational literature on mathematics highlights the importance of presenting the development of a proof attempt alongside a final, or currently accepted, proof artefact. We have developed an argumentation-based framework in which this is possible. A grounded extension of a dialogue can be produced, representing a currently accepted, collaboratively constructed, proof or theory. Since the record of the dialogue can be presented alongside this proof, the framework delivers the history of a proof attempt as well as the proof artefact. Similarly, social aspects in human mathematics have been shown to be integral to the human context. Technologies which are able to support mathematicians in the collective construction of mathematical knowledge, in a variety of ways, such as highlighting conflicting commitments or unresolved moves, finding similarities and conflicts across different discussions going on in parallel among otherwise independent groups of arguers, storing past discussions and making them searchable, and so on, are essential.

Our work develops across three arcs – a theoretical model, an abstraction level and a computational model – in order to take into account studies of human mathematical reasoning. In particular, we have:

1. taken Lakatos's informally specified technique for conducting dialogue in mathematics,
2. modeled it as a formal dialogue system,
3. expressed this model in a domain specific language for dialogue game specification, DGDL,

Table 4
Second strategy of testing the proof: countering a lemma.

Formalisation		Representation		Evaluation	
LG system	*Dialogue theory*	*AIF*	*ASPIC*$^+$	AF_{LG}	GE_{LG}
Opponent – Attack					
(1-l) *LocalCounter* (M_2, L_3) **[L2.1, S16.3b]**	L_1 L_2	$\mathbf{I}_{11}$: Take the triangular network which results from performing the first two operations on a cube. Now if I remove a triangle from the inside of this network, as one might take a piece of the jigsaw puzzle, I remove one triangle without removing a single edge or vertex $[M_2]$ $\mathbf{CA}_8$: $(\mathrm{I}_4, \mathrm{I}_{11})$ $[(L_3, M_2)]$ $\mathbf{CA}_9$: $(\mathrm{I}_{11}, \mathrm{I}_4)$ $[(M_2, L_3)]$	$\mathcal{K} = \{C, L_1, L_2, L_3, M_2\}$ $\mathcal{R}_{LG} = \{(L_1, L_2, L_3) \Rightarrow C\}$ $^- = \{(L_3, M_2), (M_2, L_3)\}$	$AR_{LG} = \{L_1, L_2, L_3, \omega, M_2\}$ $att_{LG} = \{(M_2, \omega), (M_2, L_3), (L_3, M_2)\}$	L_1 L_2
Proponent – Defence					
(1-m) *LocalLemmaInc* (M_2, L_3, K) **[L4.1, S6.1]**	L_1 L_2 K	$\mathbf{I}_{12}$: if we drop the boundary triangles one by one from a triangulated map, there are only two alternatives – the disappearance of one edge or else of two edges and a vertex $[K]$ $\mathbf{MA}_1$: $(\{\mathrm{I}_4, \mathrm{I}_{11}\}, \mathrm{I}_{12})$ $[(\{L_3, M_2\}, K)]$	$\mathcal{K} = \{C, L_1, L_2, L_3, M_2, K\}$ $\mathcal{R}_{LG} = \{(L_1, L_2, L_3) \Rightarrow C\}$ $^- = \{(L_3, M_2), (M_2, L_3)\}$	$AR_{LG} = \{L_1, L_2, L_3, \omega, M_2, K\}$ $att_{LG} = \{(M_2, \omega), (M_2, L_3), (L_3, M_2)\}$	L_1 L_2 K
(1-n) *ProofDone*() **[L1.3, S19]**	L_1 L_2 K L_1, L_2, K so C	$\mathbf{RA}_6$: $(\{\mathrm{I}_2, \mathrm{I}_3, \mathrm{I}_{12}\}, \mathrm{I}_1)$ $[(\{L_1, L_2, K\}, C)]$	$\mathcal{K} = \{C, L_1, L_2, L_3, M_2, K\}$ $\mathcal{R}_{LG} = \{(L_1, L_2, L_3) \Rightarrow C, (L_1, L_2, K) \Rightarrow C\}$ $^- = \{(L_3, M_2), (M_2, L_3)\}$	$AR_{LG} = \{L_1, L_2, L_3, \omega, M_2, K, \omega'\}$ $att_{LG} = \{(M_2, \omega), (M_2, L_3), (L_3, M_2)\}$	L_1 L_2 K ω'

Table 5
The final grounded extension at the end of the example dialogue.

Argument	Text
L_1	Any polyhedron, after having a face removed, can be stretched flat on the blackboard
L_2	in triangulating a map, one will always get a new face for any new edge
M_1	for the hollow cube $\mathrm{V} - \mathrm{E} + \mathrm{F} = 4$
D	A polyhedron is a surface consisting of a system of polygons
δ	[D so This pair of nested cubes is not a polyhedron at all]
K	if we drop the boundary triangles one by one from a triangulated map, there are only two alternatives – the disappearance of one edge or else of two edges and a vertex
$\omega\prime$	[$\{L_1, L_2, K\}$ so for all polyhedra, $\mathrm{V} - \mathrm{E} + \mathrm{F} = 2$]

4. defined operational semantics for the specification in terms of updates to argumentation structures expressed in a natural language dialogue for argumentation, AIF,
5. induced abstract argumentation frameworks from AIF via the structured argumentation system *ASPIC*$^+$
6. shown the consequences of AIF updates at the abstract layer
7. demonstrated how those abstract semantics yield a grounded extension that provably always corresponds to the theory that has been collaboratively created by the dialogue participants
8. shown how the entire pipeline can be implemented and then skinned with the Arvina interface (using TOAST, DungO-Matic, AIFdb and DGEP)
9. and, finally, shown how an example of Lakatos's informal logic of mathematical discovery can be formalised.

This is the first time that formally specified and fully implemented argumentation tools right through the abstraction hierarchy from natural language dialogue through structured argumentation to instantiated abstract argumentation have been brought together and applied to a specific, demanding domain of human reasoning. The foundation that has been laid here allows new explorations into mixed-initiative, collaborative reasoning between human and software participants in interactions which are both naturalistic but formally constrained and well-defined, with the potential to impact both the pedagogy and the professional practice of mathematics.

We emphasise that no formal linguistic analysis or NLP has been incorporated into the current effort. Such an effort must be reserved for future work, and it would play a crucial role in an automated mathematical dialogue system. Contemporary strategies from natural language processing, including the field of argument mining [73] have gained traction within the broader field of discourse mining and could help inform future NLP-based efforts in this direction. Progress has made

in understanding the language that mathematicians use [38,28], integrating this with reasoning systems, and generating human-like output [39].

Representation and reasoning are two important and intertwined branches of AI. Our paper has primarily focused on representation *of* reasoning. Moreover, this is not reasoning by way of standard deductive logic that is often used in logical foundations of mathematics, but is rather, per Lakatos, a closer approximation of the kind of reasoning used in everyday mathematics. Appendix A illustrates that our system models, with reasonable fidelity, the key processes that take place in mathematical dialogues leading to proofs. This can also be taken as a proof-of-concept that implies that the theoretical and technical achievements of the paper are on the right track.

Each of the major arcs mentioned in Section 1 could be continued in future work.

- On the theory side, expanding the range of mathematical dialogues that can be modeled would in some cases require moving beyond Lakatos's informal logic, e.g., to include problem solving and problem specification in the spirit of Pólya [76,84].
- The abstraction layer could usefully be upgraded with a richer *instantiation layer*: per [11], in contrast to abstract argumentation, instantiated argumentation makes "the internal structure and content of the arguments, the relationships between the arguments in virtue of their content, and the application of argumentation" available to reasoning. This would be relevant for modelling, e.g., the lemmas or sequences of lemmas might be attacked in terms of their structure. This work would go hand-in-hand with the introduction of Pólya-style planning and heuristics.
- On the technical side, another major challenge for future work will be the design of agents that can participate *in* Lakatosian reasoning. This has been piloted in [70] (see [71] for a summary), but the formality and precision of the current offering motivate a new approach to this challenge, and provide groundwork for further advances. For example, following [78], it would be useful if interlocutors were able to identify the relevant moves among the permitted ones.

The research described here presents a solid basis for such a programme of work, because it represents the kind of reasoning that people actually use in practice. The extent to which human reasoning will jibe with "human-oriented theorem proving" systems (as used and surveyed briefly in [39]) remains to be seen. The inferential construction of the grounded extension in our system automates some basic reasoning tasks. Extensions, restrictions, or other modifications to the Lakatosian model – to represent richer or simply different forms of reasoning – along with corresponding extensions to the backend are reasonable projects for follow-up work. For example, we recently presented work with a limited Lakatosian framework for discussions about design artefacts [27]. More broadly, representing practical reasoning bears on several of the challenges about which AI researchers had been initially optimistic [86]. The systems we work with make available inherently explicable patterns of reasoning, which stands at a contrast to some of the impressive successes in the connectionist tradition. Pragmatics are essential for realising AI that can work in dialogue with humans.

Our long term goal is to build a seamless interaction between automatic, semi-automatic and manual processes in shaping the dynamics and the outcome of argumentative interactions among multiple parties, possibly including both human users and artificial agents. Here we have developed a new approach in which tools and theories from the argumentation community can be deployed to build a bridge between interactive proof tools and human mathematicians, thus helping to close the gap between the two disparate styles of reasoning. We have focused on two key interrelated areas of human mathematics: informal proof together with corresponding representations of proofs-in-progress; and the social aspects of proof development. Our treatment of these issues here opens the door to mixed-initiative collaboration in mathematics. This will be a step towards our vision of a mathematics social machine, in which mathematicians view software systems as valued collaborators and respected fellow mathematicians.

Acknowledgements

The authors would like to thank all four reviewers for their exceptionally thorough, insightful and helpful reviews. We made substantial changes to the paper in light of their feedback and there is no doubt that the paper is much improved because of them. We further gratefully acknowledge that this work has been supported in part by the Polish National Science Centre under grant 2011/03/B/HS1/04559, in part by the EPSRC under grants EP/G060347/1, EP/N014871/1 and EP/K037293/1, and in part by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open Grant number 611553 (COINVENT).

Appendix A. Lakatos in the wild

While acknowledging Lakatos's contribution to the field, fellow philosophers of mathematical practice have criticised him for over-reliance on a few case studies, historical inaccuracy and narrowness of application of his theory [112,111,113]. This provides an impetus for a wider search for empirical examples. Here, we illustrate the feasibility of using our formalisation of Lakatosian informal mathematics to model a real-world example of mathematical dialogue.

Until recently, naturalistic observation of mathematical discussions would typically only be possible for the direct participants in those discussions. However, archives of online discussions usefully make most of the same information that participants share available to researchers [118]. Such discussions are very different from the more formal presentations

typical of research papers, which present polished proofs but lack transparency about how those proofs were obtained [114, 119].

In 2009, Timothy Gowers initiated the Polymath series of experiments in online collaborative mathematics, in which problems are posted online, and an open invitation issued for people to try to solve them collaboratively, documenting every step of the ensuing discussion [110]. The result is an unusual example of a public record of several episodes of mathematical activity that lead to a proof. Martin and Pease [115,117] discuss the implications of such crowdsourced mathematical activity for the production of mathematics. They describe preliminary investigations of the community question answering system MathOverflow [108], and the Polymath collaborations [110], and develop a detailed analysis of one 'Mini-Polymath' project for students [109], noting aspects of the discussion which followed the conversational patterns which Lakatos identified. We build on this work below.

The problem in [109], taken from the International Mathematical Olympiad, asks about a certain geometrical construction involving lines rotating around a set of points in the plane; it refers to an example from this class as a "windmill process."[18] Between the problem proposal and Terrence Tao's certification of the answer in thread 27, approximately seventy four minutes passed [117].

The following set of hand-selected excerpts are hand-coded using the formalism introduced in Section 3 to show explicitly how Lakatosian reasoning contributes to the core steps in the development of the proof. Sometimes near-repetitions appear, when two or more comments are posted simultaneously (we mark these as "Repetition of previous move" below); similarly, we expand some simple statements sometimes into a sequence of several moves in the Lakatos Game. With these clarifications in mind, the following excerpts trace the highlights of the proof, and their corresponding mapping into the dialogue system introduced in Section 3.

A.1. Thread 3: what is a line?

At the beginning of this thread early on in the discussion, Anonymous decides to simplify things, moving from the running *Conjecture*('*problem statement holds*') to a related game dealing with a similar conjecture in the safe domain of convex polygons. The discussants readily agree that the proposed *Conjecture*('*statement holds in the convex polygon case*') is true. Variations on this case are then considered as potential counterexamples to the original conjecture. These are rejected after the basic concepts in the problem statement have been clarified.

1: Anonymous: If the points form a convex polygon, it is easy.	a. P: *Conjecture*('*statement holds in the convex polygon case*'). **[L1.1, S1, C1.1]** b. P: *Lemma*('*a windmill process walks around the vertices of the convex polygon*' $L_{3.1}$) **[L1.2, S2.2, C2.1]** c. P: *ProofDone*. **[L1.3, S3.2]**
2: Thomas H: Yes. Can we do it if there is a single point not on the convex hull of the points?	d. P: *Conjecture*('*statement holds in the convex polygon plus point case*'). **[L1.1, S1, C1.1]**
3: Jerzy: Say there are four points: an equilateral triangle, and then one point in the centre of the triangle. No three points are collinear. It seems to me that the windmill can not use the centre point more than once! As soon as it hits one of the corner points, it will cycle indefinitely through the corners and never return to the centre point. I must be missing something here…	e. O: *GlobalCounter*('*equilateral triangle plus point*', '*statement holds in the convex polygon case*'). **[L2.3, S4.1]**
4: Joe: This isn't true – it will alternate between the centre and each vertex of the triangle.	f. P: *MonsterBar*('*equilateral triangle plus point*', '*statement holds in the convex polygon case*', '*line is alternating*'). **[L5.1, S5.3]**
5: Thomas H: No, you're not right. Let the corner points be A, B, C, clockwise, M the centre. If you start in M, you first hit say A, then C, then M, then B, then A.	g. P: *PDefinition*('*equilateral triangle plus point*', '*line is alternating*', '"*line" extends in both directions*'). **[L5.3, S10]**
6: Jerzy: Ohhh…I misunderstood the problem. I saw it as a half-line extending out from the last point, in which case you would get stuck on the convex hull. But apparently it means a full line, so that the next point can be "behind" the previous point. Got it.	h. O: *MonsterAccept*('*equilateral triangle plus point*', '*line is alternating*'). **[L6.1, S12.1]**

[18] "Let S be a finite set of at least two points in the plane. Assume that no three points of S are collinear. A windmill is a process that starts with a line ℓ going through a single point $P \in S$. The line rotates clockwise about the pivot P until the first time that the line meets some other point Q belonging to S. This point Q takes over as the new pivot, and the line now rotates clockwise about Q, until it next meets a point of S. This process continues indefinitely. Show that we can choose a point P in S and a line ℓ going through P such that the resulting windmill uses each point of S as a pivot infinitely many times." [109]

A.2. Thread 11: hitting all the points

In this thread, Anonymous offers a potential characterisation of generality. We again retain *Conjecture*(*'problem statement holds'*), and Anonymous first proposes *Lemma*(*'we can start with any point'*). This is refined into the claim *Lemma*(*'the line does not matter'*). However, counterexamples to this claim are produced, and Anonymous must refine the claim again.

7: Anonymous: One can start with any point (since every point of S should be pivot infinitely often), the direction of line that one starts with however matters!	a. P: *Lemma*(*'we can start with any point'* $L_{11.1}$).	**[L1.2, S2.2, C2.1]**
8: Nemanja: In other words, we can start with any point and 'just' need to choose a second point through which will we draw a line.	[Repetition of previous move.]	
9: Anonymous: Perhaps even the line does not matter! Is it possible to prove that any point and any line will do?	b. P: *Lemma*(*'the line does not matter'* $L_{11.2}$).	**[L1.2, S3.1, C2.1]**
	c. P: *ProofDone*.	**[L1.3, S3.2]**
10: Thomas H: No, if you start with two points on the convex hull (ordered in the right way) you stay on the convex hull.	d. O: *LocalCounter*(*'two points on the convex hull'*, $L_{11.2}$)	**[L2.1, S4.2]**
11: Nemanja: It is not possible, two consecutive points on convex hull will not do.	[Repetition of previous move.]	
12: Zhecka: Sure a choice of line is important. Imagine S is a set of vertices of a convex polygon P (triangle, say) plus one point inside P.	[Repetition of previous move.]	
13: Anonymous: Only the starting point matters. By the problem statement, it appears that the initial angle is irrelevant to the existence of a pivot point P^* from which all of S is traversed. Every point in S is a pivot point, but only with a specific range of starting angle (e.g. those consistent with the cycle generating S). The union of these intervals must necessarily be $[0, 2\pi)$, and thus we can assume WLOG that the starting angle is 0 (and thus we single out a specific point – or points in the case of $\|S\| = 2$).	e. P: *LocalLemmaInc*(*'two points on the convex hull'*, $L_{11.2}$, *'the initial angle is irrelevant within a specific range'*).	**[L4.1, S6.1, C2.2]**

A.3. Thread 14 (first part): splitting in two sets, conclusion

This thread, which contains the final solution to the problem, begins with Anonymous advancing *Lemma*(*'we can separate the points in two parts of roughly equal size'*). This draws some initial interest, but the discussants aren't sure how it will be useful. A considerable amount of time passes before Garf is able to make something more concrete from the earlier remarks, in the form of a sequence of lemmas that, he claims, secure the desired result. Garf's chain of reasoning will be certified as a solution to the problem only after the claims have been thoroughly vetted. Further minor criticisms and a recap of the solution are elided here.

14: Anonymous: I'm not sure but as no three points are collinear, one can always find a line which splits the points into two sets whose number of elements differ at most one?	a. P: *Lemma*(*'we can separate the points in two parts of roughly equal size'* $L_{14.1}$).	**[L1.2, S6.1, C2.1]**
15: Thomas H: That is surely true. How could this help us?	[Not coded.]	
16: Anonymous: Something like one can find this no matter how we choose the first point. Then in some time the windmill must be parallel to the line through these points. This line must be unique or else it splits the points such that number of elements differ at least two.	[Not coded.]	
17: Justin W Smith: It appears that the number of points to the "left" or "right" of the line is constant through the entire process!	[Not coded.]	
18: Garf: I think this solves the problem. Start with a line which separates the points into two parts of roughly same size (their cardinal differ by at most one, not counting the point to which the line is attached). Then run the process until the line is "upside-down", and so has turn by exactly π. Every point has gone from the right of the line to the left of the line (easy to see if the number of points is odd, you have to be a bit more crafty if it is even), and no point can go from left to right or right to left without touching the line. Add the previous remarks (the process will always come back to its initial configuration), and every point will be visited infinitely often.	b. P: *Lemma*(*'we can run the process until the line is upside down'* $L_{14.2}$).	**[L1.2, S2.2, C2.1]**
	c. P: *Lemma*(*'every point goes from right to left'* $L_{14.3}$).	**[L1.2, S3.1, C2.1]**
	d. P: *Lemma*(*'no point can go from left to right without touching the line'* $L_{14.4}$).	**[L1.2, S3.1, C2.1]**
	e. P: *Lemma*(*'the process will return to its initial configuration'* $L_{14.5}$).	**[L1.2, S3.1, C2.1]**
	f. P: *Lemma*(*'each point will be visited infinitely often'* $L_{14.6}$).	**[L1.2, S3.1, C2.1]**
	g. P: *ProofDone*.	**[L1.3, S3.2]**

19: Gal: Very nice! Don't we run into problems with a convex hull though? Take a square with a point in the middle (M) and pass the diagonal of the square (not through M) – it seems to me M is never visited (though I may be wrong here). I think we should be more specific in our initial choice of line, maybe?	h. O: *HybridCounter*('*point inside square*', '*problem statement holds*', $L_{14.6}$). **[L2.2, S4.3]**
20: Gal: No. This example is false :)	i. P: *HybridLemmaInc*('*point inside square*', $L_{14.6}$,). **[L4.2, S7.1, C2.3]** j. P: *Conjecture*('*problem statement holds, even for point inside square*'). **[L4.2, S18, C1.1]** k. P: *ProofDone*. **[L1.3, S2.1a]**
21: Zhecka: Yes, it seems to be a correct solution!	[Repetition of previous moves.]
22: Gal: This seems to be right, but there something I don't understand. Please see if you can help me with it: Start with a square and a point inside it (M): start with a tangent to the square (your solution demands a more equal division of points, I know). When we get to the opposite vertex of the square all points moved from one side of the line to the other, but not all points have been visited (M will never be visited). The argument is almost exactly the same, so it seems that the equal division of points plays a crucial role, but I don't understand what role exactly. Can we pin it down precisely?	l. O: *LocalCounter*('*M is never visited*', $L_{14.4}$). **[L2.1, S4.2]**
23: Garf: If I understand well your example: the problem is that you must give an orientation to the line. Then, left and right are defined with respect to this orientation: if the line has made half a turn, then left and right are reversed. In your example, I think most of the points move from, say, the part at the top of the line to the part at the bottom of the line, but always stay at the right of the line.	m. P: *LocalLemmaInc*('*M is never visited*', $L_{14.4}$, '*left and right must be reversed after a rotation*'). **[L4.1, S6.1, C2.2]**
Gal: Got it! Kind of like a turn number in topology. Thanks! :)	[Not coded.]

A.4. Summary

These examples show that Lakatos-style reasoning can be used to describe real world examples of mathematical conversations. It is also clear that there are reasoning steps involved that are not classically Lakatosian, for instance surrounding the clarification of terminology rather than concepts (in Thread 3 (g–h)). The relationship between the overall game and a sub-game like the one in Thread 3 (a–c) is not formally specified in our protocol, although it would not be difficult to do so [116]. In some cases, a more detailed analysis might yield further insights into the discussants' thought process. For example, consider Turns 14–17 in Thread 14, which comprise a more complex exchange than would be needed if the only goal was to introduce a lemma. Nevertheless, the overall structure of the discussion is coherent with the Lakatosian framework.

References

[1] A. Aberdein, Persuasive definition, in: H.V. Hansen, C.W. Tindale, A.V. Colman (Eds.), Argumentation and Rhetoric, Vale Press, Newport News, VA, 1998.
[2] A. Aberdein, The uses of argument in mathematics, Argumentation 19 (2005) 287–301.
[3] A. Aberdein, Managing informal mathematical knowledge: techniques from informal logic, in: J.M. Borwein, W.M. Farmer (Eds.), MKM 2006, in: LNAI, vol. 4108, Springer-Verlag, Berlin, 2006, pp. 208–221.
[4] A. Aberdein, The informal logic of mathematical proof, in: B.v. Kerkhove, J.P.v. Bendegem (Eds.), Perspectives on Mathematical Practices: Bringing Together Philosophy of Mathematics, Sociology of Mathematics, and Mathematics Education, in: Logic, Epistemology, and the Unity of Science, vol. 5, Springer, 2007, pp. 135–151.
[5] A. Aberdein, Mathematics and argumentation, Found. Sci. 14 (1–2) (2009) 1–8.
[6] A. Aberdein, The dialectical tier of mathematical proof, in: Argumentation: Cognition & Community, 2011.
[7] A. Aberdein, I.J.E. Dove, The Argument of Mathematics, Logic, Epistemology, and the Unity of Science Series, vol. 30, Springer, Netherlands, 2013.
[8] J. Alcolea Banegas, L'argumentació en matemàtiques, in: E. Casaban i Moya (Ed.), XIIè Congrés Valencià de Filosofia, Valencià, 1998, pp. 135–147.
[9] M. Aschbacher, Highly complex proofs and implications of such proofs, in: A. Bundy, M. Atiyah, A. Macintyre, D. MacKenzie (Eds.), Phil. Trans. of the Royal Society, vol. 363, 2005, pp. 2401–2406.
[10] K. Atkinson, T. Bench-Capon, P. McBurney, A dialogue game protocol for multi-agent argument over proposals for action, Auton. Agents Multi-Agent Syst. 11 (2) (2005) 153–171.
[11] K. Atkinson, A. Wyner, The value of values in computational argumentation, in: K. Atkinson, H. Prakken, A. Wyner (Eds.), From Knowledge Representation to Argumentation in AI, Law and Policy Making: A Festschrift in Honour of Trevor Bench-Capon on the Occasion of His 60th Birthday, College Publications, 2013, pp. 39–62.
[12] T.J. Bench-Capon, P.E. Dunne, Argumentation in artificial intelligence, Artif. Intell. 171 (10–15) (2007) 619–641.
[13] T. Berners-Lee, M. Fischetti, Weaving the Web – the Original Design and Ultimate Destiny of the World Wide Web by Its Inventor, HarperBusiness, 2000.
[14] F. Bex, J. Lawrence, C. Reed, Generalising argument dialogue with the dialogue game execution platform, in: S. Parsons, N. Oren, C. Reed, F. Cerutti (Eds.), Proceedings of the Fifth International Conference on Computational Models of Argument, COMMA 2014, IOS Press, Pitlochry, 2014, pp. 141–152.
[15] F. Bex, J. Lawrence, M. Snaith, C. Reed, Implementing the argument web, Commun. ACM 56 (10) (Oct. 2013) 66–73.
[16] F. Bex, H. Prakken, S. Modgil, C. Reed, On logical reifications of the argument interchange format, J. Log. Comput. 23 (5) (2013) 951–989.
[17] E. Black, K. Atkinson, Choosing persuasive arguments for action, in: Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2011, International Foundation for Autonomous Agents and Multiagent Systems, 2011, pp. 905–912.

[18] J.C. Blanchette, C. Kaliszyk, L.C. Paulson, J. Urban, Hammering towards QED, J. Formaliz. Reason. 9 (1) (2016) 101–148.
[19] K. Budzynska, K. Debowska, Dialogues with conflict resolution: goals and effects, in: Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue, 2010, pp. 59–66.
[20] A. Bundy, The nature of mathematical proof, in: A. Bundy, M. Atiyah, A. Macintyre, D. MacKenzie (Eds.), Phil. Trans. of the Royal Society, vol. 363, 2005, pp. 2329–2461.
[21] A. Bundy, Automated theorem provers: a practical tool?, Ann. Math. Artif. Intell. 61 (2011) 3–14.
[22] H.C. Bunt, W.J. Black, Abduction, Belief, and Context in Dialogue: Studies in Computational Pragmatics, MIT Press, 2000.
[23] A.L. Cauchy, Recherches sur les polyèdres, J. Éc. Polytech. 9 (1813) 68–86.
[24] F. Cerutti, M. Giacomin, M. Vallati, Argsemsat: solving argumentation problems using SAT, in: Computational Models of Argument – Proceedings of COMMA 2014, Atholl Palace Hotel, Scottish Highlands, UK, September 9–12, 2014, 2014, pp. 455–456.
[25] C. Chesñevar, J. McGinnis, S. Modgil, I. Rahwan, C. Reed, G. Simari, M. South, G. Vreeswijk, S. Willmott, Towards an argument interchange format, Knowl. Eng. Rev. 21 (4) (2006) 293–316.
[26] H.H. Clark, E.F. Schaefer, Contributing to discourse, Cogn. Sci. 13 (2) (1989) 259–294.
[27] R. Confalonieri, J. Corneli, A. Pease, E. Plaza, M. Schorlemmer, Using argumentation to evaluate concept blends in combinatorial creativity, in: S. Colton, H. Toivonen, M. Cook, D. Ventura (Eds.), Proceedings of the Sixth International Conference on Computational Creativity, ICCC 2015, 2015.
[28] M. Cramer, Proof-checking Mathematical Texts in Controlled Natural Language, PhD thesis, Universitäts- und Landesbibliothek, Bonn, 2013.
[29] M. Cramer, B. Fisseni, P. Koepke, D. Kühlwein, B. Schröder, J. Veldman, The Naproche Project Controlled Natural Language Proof Checking of Mathematical Texts, Springer, Berlin, Heidelberg, 2010, pp. 170–186.
[30] E. Denney, J. Power, K. Tourlas, Hiproofs: a hierarchical notion of proof tree, Electron. Notes Theor. Comput. Sci. 155 (2006) 341–359.
[31] J. Devereux, C. Reed, Strategic argumentation in rigorous persuasion dialogue, in: Proceedings of ArgMAS 2009, Springer, 2009.
[32] P. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artif. Intell. 77 (2) (1995) 321–358.
[33] W. Dvorák, S.A. Gaggl, J.P. Wallner, S. Woltran, Making use of advances in answer-set programming for abstract argumentation systems, CoRR, arXiv:1108.4942 [abs], 2011.
[34] S.L. Epstein, Learning and discovery: one system's search for mathematical knowledge, Comput. Intell. 4 (1) (1988) 42–53.
[35] S.L. Epstein, Wanted: collaborative intelligence, Artif. Intell. 221 (2015) 36–45.
[36] A. Fiedler, Natural language proof explanation, in: Mechanizing Mathematical Reasoning, 2005, pp. 342–363.
[37] R. Gaizauskas, U. Kruschwitz, M. Poesio, The SENSEI project: making sense of human conversations, in: Future and Emergent Trends in Language Technology: First International Workshop, FETLT 2015, Seville, Spain, November 19–20, 2015, in: LNCS, vol. 9577, Springer, 2016, p. 10, Revised Selected Papers.
[38] M. Ganesalingam, The Language of Mathematics, Springer, 2013.
[39] M. Ganesalingam, W.T. Gowers, A fully automatic theorem prover with human-style output, J. Autom. Reason. (2016) 1–39.
[40] G. Gonthier, Advances in the Formalization of the Odd Order Theorem, Proc. of Interactive Theorem Proving, vol. 6898, 2011.
[41] W.T. Gowers, Rough structure and classification, in: GAFA (Geometric and Functional Analysis), 2000, Special volume – GAFA2000(1-0).
[42] T. Hales, et al., A revision of the proof of the Kepler conjecture, Discrete Comput. Geom. 44 (2010).
[43] C. Hamblin, Fallacies, Methuen, London, 1970.
[44] G.H. Hardy, Mathematical proof, Mind 38 (1928) 11–25.
[45] J. Harrison, Handbook of Practical Logic and Automated Reasoning, 2009.
[46] F. Hayes-Roth, Using proofs and refutations to learn from experience, in: R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), Machine Learning: An Artificial Intelligence Approach, Tioga Publishing Company, Palo Alto, CA, 1983, pp. 221–240.
[47] R. Hersh, Mathematics has a front and a back, Synthese: New Directions in the Philosophy of Mathematics 88 (2) (1991) 127–133.
[48] J.R. Hobbs, D.A. Evans, Conversation as planned behavior, Cogn. Sci. 4 (4) (1980) 349–377.
[49] N.R. Jennings, L. Moreau, D. Nicholson, S.D. Ramchurn, S. Roberts, T. Rodden, A. Rogers, Human–agent collectives, Commun. ACM 57 (12) (2014) 80–88.
[50] A. Joshi, B.H. Weber, I.A. Sag, Elements of Discourse Understanding: Proceedings of a Workshop on Computational Aspects of Linguistic Structure and Discourse Setting, 1980.
[51] E. Kamar, Y.K. Gal, B.J. Grosz, Modeling information exchange opportunities for effective human–computer teamwork, Artif. Intell. 195 (2013) 528–550.
[52] R. Koons, Defeasible reasoning, in: E.N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy, spring 2014 edition, Metaphysics Research Lab, Stanford University, 2014.
[53] E.C.W. Krabbe, Arguments, proofs, and dialogues, [7], chapter 3, pp. 31–45.
[54] I. Lakatos, Falsification and the methodology of scientific research programmes, in: I. Lakatos, A. Musgrave (Eds.), Criticism and the Growth of Knowledge, CUP, Cambridge, 1970, pp. 91–195.
[55] I. Lakatos, Proofs and Refutations, Cambridge University Press, Cambridge, 1976.
[56] J. Lawrence, F. Bex, C. Reed, Dialogues on the argument web: mixed initiative argumentation with Arvina, in: Proceedings of the 4th International Conference on Computational Models of Argument, COMMA 2012, IOS Press, Vienna, 2012, pp. 513–514.
[57] J. Lawrence, F. Bex, C. Reed, M. Snaith, AIFdb: Infrastructure for the argument web, in: Proceedings of the Fourth International Conference on Computational Models of Argument, COMMA 2012, 2012, pp. 515–516.
[58] D.B. Lenat, J.S. Brown, Why AM and EURISKO appear to work, Artif. Intell. 23 (3) (1984) 269–294.
[59] K. Lorenz, P. Lorenzen, Dialogische Logik, WBG, Darmstadt, 1978.
[60] D. MacKenzie, Mechanizing Proof: Computing, Risk, and Trust, MIT Press, Cambridge, MA, 2001.
[61] D. MacKenzie, Computing and the cultures of proving, in: A. Bundy, M. Atiyah, A. Macintyre, D. MacKenzie (Eds.), Phil. Trans. of the Royal Society, vol. 363, 2005, pp. 2335–2350.
[62] J. Mackenzie, Four dialogue systems, Stud. Log. 51 (1990) 567–583.
[63] U. Martin, A. Pease, Mathematical practice, crowdsourcing, and social machines, in: J. Carette, D. Aspinall, C. Lange, P. Sojka, W. Windsteiger (Eds.), Intelligent Computer Mathematics, in: Lecture Notes in Computer Science, vol. 7961, Springer, Berlin, Heidelberg, 2013, pp. 98–119.
[64] P. McBurney, S. Parsons, Games that agents play: a formal framework for dialogues between autonomous agents, J. Log. Lang. Inf. (13) (2002) 315–343.
[65] P. McBurney, S. Parsons, Dialogue game protocols, in: Communication in Multiagent Systems, Springer, 2003, pp. 269–283.
[66] S. Modgil, H. Prakken, The ASPIC+ framework for structured argumentation: a tutorial, Argument Comput. 5 (1) (2014) 31–62.
[67] P. Ernest, Opening the mathematics text: what does it say?, in: E. de Freitas, K. Nolan (Eds.), Opening the Research Text: Critical Insights and In(ter)ventions into Mathematics Education, Springer, Dordrecht, 2008, pp. 65–80.
[68] S. Parsons, M. Wooldridge, L. Amgoud, An analysis of formal interagent dialogues, in: Proc. of AAMAS, 2002, pp. 394–401.
[69] A. Paseau, What's the point of complete rigour?, Mind 125 (497) (2016) 177–207.
[70] A. Pease, A Computational Model of Lakatos-style Reasoning, PhD thesis, University of Edinburgh, 2007.
[71] A. Pease, A computational model of Lakatos-style reasoning, Philos. Math. Educ. J. (ISSN 1465-2978) 27 (April 2013) (Online).

[72] A. Pease, A. Aberdein, Five theories of reasoning: inter-connections and applications to mathematics, Log. Log. Philos. 20 (1–2) (2011) 7–57.
[73] A. Peldszus, M. Stede, From argument diagrams to argumentation mining in texts: a survey, Int. J. Cog. Inform. Nat. Intell. 7 (1) (2013) 1–31.
[74] A.-V. Pietarinen, Multi-agent systems and game theory—a peircean manifesto, Int. J. Gen. Syst. 33 (4) (2004) 395–414.
[75] G. Pólya, How to Solve It, Princeton University Press, 1945.
[76] G. Pólya, Mathematical Discovery: On Understanding, Learning, and Teaching Problem Solving, John Wiley & Sons, 1981.
[77] K.R. Popper, Objective Knowledge, OUP, Ely House, London, 1972.
[78] H. Prakken, Coherence and flexibility in dialogue games for argumentation, J. Log. Comput. 15 (6) (2005) 1009–1040.
[79] H. Prakken, Formal systems for persuasion dialogue, Knowl. Eng. Rev. 21 (2006) 163–188.
[80] H. Prakken, An abstract framework for argumentation with structured arguments, Argument Comput. 1 (2010).
[81] H. Prakken, A.Z. Wyner, T.J. Bench-Capon, K.D. Atkinson, A formalisation of argumentation schemes for case-based reasoning in ASPIC+, J. Log. Comput. (2013) 1141–1166.
[82] I. Rahwan, C. Reed, The argument interchange format, in: I. Rahwan, G. Simari (Eds.), Argumentation in Artificial Intelligence, Springer, 2009.
[83] I. Rahwan, F. Zablith, C. Reed, Laying the foundations for a world wide argument web, Artif. Intell. 171 (2007) 897–921.
[84] A.H. Schoenfeld, Mathematical Problem Solving, Elsevier, 2014.
[85] J. Searle, Speech Acts: An Essay in the Philosophy of Language, Cambridge University Press, 1969.
[86] H.A. Simon, A. Newell, Heuristic problem solving: the next advance in operations research, Oper. Res. 6 (1) (1958) 1–10.
[87] A. Sloman, The well-designed young mathematician, Artif. Intell. 172 (18) (December 2008) 2015–2034.
[88] M. Snaith, J. Devereux, J. Lawrence, C. Reed, Pipelining argumentation technologies, in: P. Baroni, F. Cerutti, M. Giacomin, Simari (Eds.), Proceedings of the 3rd International Conference on Computational Models of Argument, COMMA 2010, IOS Press, 2010, pp. 447–454.
[89] M. Snaith, J. Lawrence, C. Reed, Mixed initiative argument in public deliberation, in: F. De Cindio, A. Macintosh, C. Peraboni (Eds.), From e-Participation to Online Deliberation, Proceedings of the Fourth International Conference on Online Deliberation, OD2010, Leeds, UK, 2010, pp. 2–13.
[90] M. Snaith, C. Reed, TOAST: online ASPIC+ implementation, in: Proceedings of the 4th International Conference on Computational Models of Argument, COMMA 2012, IOS Press, Vienna, 2012.
[91] M. Stone, Abduction, Belief and Context in Dialogue: Studies in Computational Pragmatics Harry Bunt and William Black (editors) (Tilburg University and UMIST) Amsterdam: John Benjamins (Natural language processing series, edited by Ruslan Mitkov, volume 1), 2000, Comput. Linguist. 28 (1) (Mar. 2002) 96–98.
[92] G. Sutcliffe, SRASS: a semantic relevance axiom selection system, in: Proceedings of ARW'07, 2007.
[93] M. Thimm, Tweety – a comprehensive collection of Java libraries for logical aspects of artificial intelligence and knowledge representation, in: Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning, KR'14, July 2014.
[94] W. Thurston, On proof and progress in mathematics, Bull., New Ser., Am. Math. Soc. 30 (2) (1994) 161–177.
[95] S. Toulmin, R. Rieke, A. Janik, An Introduction to Reasoning, Macmillan, London, 1979.
[96] D. Traum, S. Larsson, The information state approach to dialogue management, in: J. Kuppevelt, R. Smith (Eds.), Current and New Directions in Discourse and Dialogue, 2003, pp. 325–353.
[97] D.R. Traum, Computational models of grounding in collaborative systems, in: S.E. Brennan, A. Giboin, D. Traum (Eds.), Psychological Models of Communication in Collaborative Systems – Papers from the AAAI Fall Symposium, 1999, pp. 124–131.
[98] A. Turing, On Computable Numbers, with an Application to the Entscheidungs problem, Proc. Lond. Math. Soc. 42 (2) (1937) 23065.
[99] G. Vreeswijk, Reasoning with defeasible arguments: examples and applications, in: European Workshop on Logics in Artificial Intelligence, Springer, 1992, pp. 189–211.
[100] D. Walton, Argumentation Schemes for Presumptive Reasoning, Lawrence Erlbaum Associates, 1996.
[101] D. Walton, C. Reed, F. Macagno, Argumentation Schemes, Cambridge University Press, 2009.
[102] D.N. Walton, Logical Dialogue—Games and Fallacies, University Press of America, Lanham, Maryland, 1996.
[103] D.N. Walton, E.C.W. Krabbe, Commitment in Dialogue, SUNY Press, 1995.
[104] E. Weigand, Language as Dialogue: From Rules to Principles of Probability, John Benjamins Publishing, 2009.
[105] S. Wells, C. Reed, A domain specific language for describing diverse systems of dialogue, J. Appl. Log. 10 (4) (2012) 309–329.
[106] R. Wilder, Evolution of Mathematical Concepts, John Wiley and Sons, Inc., NY, 1968.
[107] A. Wyner, T. van Engers, A. Hunter, Working on the argument pipeline: through flow issues between natural language argument, instantiated arguments, and argumentation frameworks, in: Argument & Computation, vol. 7, IOS Press, 2016, pp. 69–89.

References of Appendix A

[108] Mathoverflow, http://mathoverflow.net.
[109] Minipolymath3 project, http://polymathprojects.org/2011/07/19/minipolymath3-project-2011-imo/.
[110] The polymath blog, http://polymathprojects.org/.
[111] D. Corfield, Assaying Lakatos's philosophy of mathematics, Stud. Hist. Philos. Sci. 28 (1) (1997) 99–121.
[112] P. Ernest, Social Constructivism as a Philosophy of Mathematics, State University of New York Press, Albany, NY, 1997.
[113] S. Feferman, The logic of mathematical discovery vs. the logical structure of mathematics, in: P.D. Asquith, I. Hacking (Eds.), Proceedings of the 1978 Biennial Meeting of the Philosophy of Science Association, vol. 2, Philosophy of Science Association, East Lansing, Michigan, 1978, pp. 309–327.
[114] R. Hersh, Mathematics has a front and a back, Synthese 88 (2) (1991) 127–133.
[115] U. Martin, A. Pease, Mathematical practice, crowdsourcing, and social machines, in: J. Carette, D. Aspinall, C. Lange, P. Sojka, W. Windsteiger (Eds.), Intelligent Computer Mathematics, in: Lecture Notes in Computer Science, vol. 7961, Springer, Berlin, Heidelberg, 2013, pp. 98–119.
[116] P. McBurney, S. Parsons, Games that agents play: a formal framework for dialogues between autonomous agents, J. Log. Lang. Inf. 11 (3) (2002) 315–334.
[117] A. Pease, U. Martin, Seventy four minutes of mathematics: an analysis of the third mini-polymath project, in: Proc. of the AISB Symp. on Mathematical Practice and Cognition II, 2012, pp. 19–29.
[118] G. Stahl, Studying Virtual Math Teams, 1st edition, Springer, Sep. 2010.
[119] W. Thurston, On proof and progress in mathematics, Bull., New Ser., Am. Math. Soc. 30 (2) (1994) 161–177.