Playing Lorenzen Dialogue Games on the Web*

Jesse Alama[1] and Sara L. Uckelman[2]

[1] Center for Artificial Intelligence
New University of Lisbon
j.alama@fct.unl.pt

[2] Institute for Logic, Language, and Computation
Universiteit van Amsterdam
S.L.Uckelman@uva.nl

Abstract

We announce an interactive website for exploring logic with the help of Lorenzen dialogue games. The site allows one to play dialogue games and computes winning plays and winning strategies from initial segments of such games. A variety of dialogue rule sets are available, allowing one to investigate different logics. We have also implemented several formula translations, so that one can learn how dialogue games vary as one changes their initial formulas. We discuss some heuristics for computing winning plays and winning strategies.

1 Introduction

Lorenzen dialogues [8, 6] are two-player logic games [5] in which a Proponent starts by laying down a logical formula that the Opponent then attacks; the game proceeds in alternating turns by breaking down the formulas according to their main connectives, or by returning to previously asserted formulas. Each player aims to force the other into a position from which no more moves can be made. The structure and outcome of the game is intended to say something about the logical content of the formula with which it started.

These games can be played with a computer, and indeed doing so is valuable because it can help ensure that one is following the rules, which, for some Lorenzen dialogue games, are not without some measure of arbitrariness and can be unintuitive. Moreover, a computer-based approach helps one to acquire more experience with dialogue games than would be possible otherwise, thereby giving the dialogue game enthusiast a deeper exposure to the subject.

The purpose of this paper is to announce an interactive website for playing Lorenzen dialogue games and to discuss some of its principle features. The site allows one to specify a propositional signature and an initial formula in that signature, to play the game as either Proponent or Opponent, to see what game rules apply (and how the ones that do not apply fail), and to edit the rules in a rudimentary way, thereby allowing one to get a better sense for what rules play which logical roles.

Our work was inspired by the DiaLog system [3, 2], developed in the late 1990s in Erlangen by J. Ehrensberger. DiaLog is written in the Scheme programming language and has a graphical interface using the Tk framework. More than providing an attractive interface for playing dialogue games, DiaLog's main advantage is that it supports investigations of *strategies* (as well as paying particular games). DiaLog is a well-polished, professional system and is even used

*Both authors were funded by the FCT/NWO/DFG project "Dialogical Foundations of Semantics" (DiFoS) in the ESF EuroCoRes programme LogICCC (FCT LogICCC/0001/2007; LogICCC-FP004; DN 231-80-002; CN 2008/08314/GW). An earlier version of this note was presented in Mathematical User-Interfaces Workshop 2010 at the Ninth Mathematical Knowledge Management conference, Paris, July 2010.

Assertion	**Attack**	**Response**
$\varphi \wedge \psi$	$\wedge_L$	φ
	$\wedge_R$	ψ
$\varphi \vee \psi$	?	φ or ψ
$\varphi \to \psi$	φ	ψ
$\neg\varphi$	φ	—

Table 1: Particle rules for dialogue games

for teaching logic in a classroom setting in Erlangen. The principal difference between DiaLog and our system is that our work is web-based and thus requires no custom software beyond a standard web browser.

The site is live at `http://dialogical-logic.info/`. It is implemented with the UnCommon Web framework for building dynamic websites in the Common Lisp programming language [11]. The Common Lisp Object System (CLOS) is employed throughout. A Lisp-based solution allows for considerable flexibility in both developing and maintaining the system. It is our hope that by putting the system on the web, we can attract a wide audience to dialogue games.

2 Dialogue games

The basis of the dialogical approach to logic is finitary open two-person zero-sum games between the players Proponent, P and Opponent, O [7]. For a set S of dialogue rules governing how the game is to proceed, a formula φ is S-valid iff Proponent has a winning strategy for φ. The dialogical rules are divided into two types: structural, and particle. Particle rules are governed by the logical connectives; they explain how formulas can be attacked and defended by the two players depending only on the structure of the formula. Structural rules restrict the possible moves of the players at any given round of the dialogue without necessarily being determined by the structure of any formula [4]. In addition to propositional formulas, there are the three so-called *symbolic attack* expressions, ?, $\wedge_L$, and $\wedge_R$. The standard particle rules for the basic propositional language are given in Table 1. Note that there is no way to defend against an attack against a negation; the only appropriate "defense" against an attack on a negation $\neg\varphi$ is to continue the game with the new information φ. There is a wide range of possibility for the structural rules. Some commonly used structural rules are the following [4] (let X and Y range over P and O):

(D10) P may assert an atomic formula only after it has been asserted by O before.

(D11) If p is an P-position, and if at round $n-1$ there are several open attacks made by O, then only the latest of them may be answered at n (and the same with P and O reversed).

(D12) An attack may be answered at most once.

(D13) A P-assertion may be attacked at most once.

(E) O can react only upon the immediately preceding P-statement.

The rulesets $\mathrm{D} \coloneqq \{\mathrm{D10}, \mathrm{D11}, \mathrm{D12}, \mathrm{D13}\}$ and $\mathrm{E} \coloneqq \mathrm{D} \cup \{\mathrm{E}\}$ (abusing notation by letting "E" refer both to the ruleset as well as its characteristic rule) corresponds to intuitionistic logic (φ is D-valid or E-valid iff φ is intuitionistically valid); the ruleset $\mathrm{CL} \coloneqq \{\mathrm{D10}, \mathrm{D13}, \mathrm{E}\}$ corresponds to classical logic.

The game begins with P laying down some initial formula, which O then attacks. It may come to pass at some point in the game that no more moves can be made, or it may be that the players can continue the game without stopping. Naturally, the structure of the game and the possibilities for its development depend on the set S of structural rules. The following definitions capture these notions.

Definition 2.1. Given a set S of structural rules, an *S-dialogue* for a formula φ is a dialogue commencing with φ that adheres to the particle rules and the rules of S. Proponent *wins* an S-dialogue if there is a round where Proponent has moved and there is no move that Opponent can legally make. A *branch* of a rooted tree is a maximal totally ordered set of nodes that includes the root. The *S-dialogue tree $T_{S,\varphi}$* for a rule set S and a formula φ is the rooted tree whose branches are all possible ways that an S-dialogue game starting with φ can develop. An *S-winning play* for P and φ is a branch of $T_{S,\varphi}$ that ends with a win for P. An *S-winning strategy* for P and φ is a rooted subtree s of $T_{S,\varphi}$ satisfying the conditions:

1. The root of s is the root of $T_{S,\varphi}$ [the strategy begins as the game does];
2. Every branch of s is an S-dialogue won by P [P always wins];
3. If k is odd and a is a depth-k node of s (so that a corresponds to a move for P), then a has exactly one child [P always has a unique response];
4. If k is even and a is a depth-k node of s (so that a corresponds to a move for O), then a has the same children in s as it has in $T_{S,\varphi}$ [all possible moves for O at stage a are accounted for].

3 Heuristics search for wining plays and winning strategies

Using our site, one can play actual games, seeing either all options for both players, or choosing a side (Proponent or Opponent) and playing against a competitor whose moves are random. A further method for interacting with the site is by exploring *winning plays* and *winning strategies*.

When playing as both players, the site offers the user a choice from all available next moves in the game. The set of all next moves in the initial dialogue game is calculated by a simple brute-force procedure: Select all 4-tuples $\langle \text{player}, \text{statement}, \text{stance}, \text{reference} \rangle$, where

1. player $\in \{\text{Proponent}, \text{Opponent}\}$,
2. statement $\in \text{Sub}(\varphi) \cup \{?, \wedge_L, \wedge_R\}$, where φ is the initial formula of the game and $\text{Sub}(\varphi)$ is the set of all subformulas of φ,[1]
3. stance $\in \{\text{attack}, \text{defend}\}$,
4. $0 \leq \text{reference} < L$, where L is the length of the game so far

that satisfy all dialogue rules. The first three conditions do not vary as the game develops, but the range of moves mentioned in the fourth condition grows monotonically, so that calculating the set of all available moves as the game develops requires an increasing amount of computational resources. The cost of evaluating dialogues becomes apparent as one searches for winning plays, and it becomes especially acute when searching for winning strategies.

[1]This condition holds only for those rule sets adherence to which implies that only subformulas of the initial formula can legally appear. All currently implemented rule sets do have that property, but if we wished to work with dialogues for which new formulas can be brought into the game, such as material dialogues [9], we would need to modify this condition.

We now discuss some heuristics for making the search for winning plays and winning strategies more efficient. The heuristics are of two kinds: *best-first* and *fail-first*. The former kind of heuristics are tuned for making the search terminate quickly with success (if indeed success can be had early on), whereas the latter kind of heuristics are designed to terminate the search on branches where we know (or have reason to believe) that a solution cannot be found.

Since the site is designed to deal with a wide range of possible rule sets, and since the range of possible rules gives rise to such a wide variety of ways that games can develop, generic heuristics that apply to arbitrary rule sets will, generally, be weak. Here is a reasonable general heuristic:

Heuristic 1. Define a *repetition* as a move that asserts the same formula, takes the same stance, and has the same reference as some move appearing earlier.

(Weak) When searching for winning plays or winning strategies, disprefer expanding nodes corresponding to *repetitions*.

(Strong) When searching for winning plays or winning strategies, do not consider nodes corresponding to *repetitions*.

Justification: Our experience with a variety of rule sets suggests that repetitions are correlated with non-wins (either in the sense of winning play or winning strategy) because they "stall" the game without meaningfully advancing it. We know of no non-*ad hoc* rule set where repetitions are valuable.

Conjecture. *We conjecture that if φ is intuitionistically valid, then every winning* D- *or* E-*strategy for φ is non-repeating, in the sense that no move by either of the players is a repetition, and that if φ is classically valid, then every winning* CL-*strategy for φ is non-repeating.*[2]

For specific rule sets, there heuristics available depending on whether one is interested in the existence of a winning play from a certain initial sequence of plays or the existence of a winning strategy for which the initial sequence of plays forms an initial segment. We now describe a few such heuristics.

Heuristic 2. For any rule set that contains rule D10, when searching for a winning play from an initial sequence of moves, prefer expanding search tree nodes corresponding to assertions by O of an atomic formula.

Justification: Intuitively, when O concedes an atomic formula, P can exploit this information to a greater degree than when O asserts a non-atomic formula, because when D10 is present P's assertions of atomic formulas are constrained by O's.

Heuristic 2 expresses a ranking among search nodes; there could very well be cases where it is to P's advantage to defer the exposure of an atom, but our experience with various rule sets suggests that it is often to P's advantage to try to expose atoms, when possible.

Heuristic 3. When rule D13 is present and searching for a winning strategy for P, prefer expanding nodes that represent defensive moves by O.

Justification: Since D13 rules out repeat attacks by O, attacks for O should be made only when necessary. If we wish to fail quickly in our search for a winning strategy, then we should give O the greatest amount of freedom.

For the rule set $\mathrm{N} = \mathrm{D10} + \mathrm{D13} = \mathrm{CL} - \mathrm{E}$ [1], we have the following heuristic:

[2]We do not claim that if φ is intuitionistically valid, then every *branch* of the dialogue tree for φ is non-repeating. It turns out that law of excluded middle, under the Gödel-Gentzen translation, is intuitionistically valid but the dialogue tree for this formula has branches with repetitions.

Heuristic 4. When searching for winning strategies, do not consider search tree nodes that represent defensive moves made by O.
Justification: N has the curious property [1, Corollary 1] that once O defends, the same defense can be repeated by O at any later point in the game, which rules out the existence of a winning strategy.

To some extent, these heuristics conflict with each other. For example, Heuristic 1 and Heuristic 3 conflict when O repeats some earlier defense. (In fact, this is precisely the phenomenon that Heuristic 4 rules out.) Thus, Heuristic 1 is a best-first heuristic because it helps to focus the search for winning strategy when various options for P are to be considered. Heuristic 2 is also a best-first heuristic, because it aims to put P in the best possible situation when searching for some winning play (though it is not especially helpful when searching for a winning *strategy*, since all possible moves by O have to be considered anyway, not just the ones where O asserts an atom). Heuristic 4 is a fail-first heuristic, because it terminates the search the moment a defense by O is made. Likewise, Heuristic 3 is a fail-first heuristic, because it benefits O.

It is natural to ask how these heuristics perform. Do they, or can they, amount to efficient decision procedures for the logics that they generate? For the rule sets D and E, which capture intuitionistic logic, an interesting puzzle is raised. Propositional intuitionistic logic is decidable: There is a decision procedure that, given a propositional formula φ, determines whether φ is intuitionistically valid. How could one decide, given φ, whether $\vDash_{\mathsf{E}} \varphi$? One argument [10] invokes the subformula property for intuitionistic sequent calculus derivations (which is proved with the help of cut elimination) and thus bounds the depth of the search tree. Although the development of an intuitionistic dialogue game in accordance with the particle rules has the flavor of a sequent calculus derivation that adheres to the subformula property, there is a notable difference: Proponent can, in general, repeat earlier moves (though Opponent is somewhat more constrained). We conjectured earlier that if φ is intuitionistically valid, then every winning strategy for φ is non-repeating, in the sense that no move by either of the players gets repeated. The difficulty is to deal with the case where φ is *not* intuitionistically valid, because, since there is no winning strategy for φ, there are, in general, many infinite branches (though of course some dialogue trees for intuitionistically invalid formulas are finite); we require a criterion by which we can rule out further development of the dialogue tree (which generally grows exponentially). In the case of sequent calculus, this criterion comes from the subformula property; we lack an analogous criterion for dialogue games.

4 Future work

The site currently deals only with propositional logic, but we plan to extend it to deal with full first-order logic. We also aim to extend our work to other logics, such as modal and linear logic.

As it stands, the procedure for evaluating moves relative to a dialogue rule set is naïve and inefficient. We intend to explore the possibility of viewing the evaluation problem as a constraint satisfaction problem, wherein dialogue rules are recast as constraints; an efficient constraint solver could then be used. Using techniques from constraint satisfaction, we might also discover *implied* constraints (consequences of the rules, qua constraints) that could improve performance even further.

Finally, we aim to implement a mechanism for exploring the equivalence of sequent calculus derivations and winning strategies for the rule sets D and E , as well as a method to translate between D- and E-strategies.

References

[1] J. Alama and S. L. Uckelman. A curious dialogical logic and its composition problem, 2010. preprint, `http://arxiv.org/abs/1008.0080`.

[2] DiaLog: A system for dialogue logic. `http://www8.informatik.uni-erlangen.de/IMMD8/staff/Goerz/DiaLogic/index.html`.

[3] J. Ehrensburger and C. Zinn. DiaLog: A system for dialogue logic. In W. McCune, editor, *Automated Deduction, Proceedings of CADE-14*, pages 446–460. Springer, 1997.

[4] W. Felscher. Dialogues, strategies, and intuitionistic provability. *Annals of Pure and Applied Logic*, 28:217–254, 1985.

[5] W. Hodges. Logic and games. In E. N. Zalta, editor, *Stanford Encyclopedia of Philosophy*. Spring edition, 2009. `http://plato.stanford.edu/archives/spr2009/entries/logic-games/`.

[6] L. Keiff. Dialogical logic. In E. N. Zalta, editor, *Stanford Encyclopedia of Philosophy*. Summer edition, 2009. `http://plato.stanford.edu/archives/sum2009/entries/logic-dialogical/`.

[7] K. Lorenz. Basic objectives of dialogue logic in historical perspective. In S. Rahman and H. Rückert, editors, *New Perspectives in Dialogical Logic*, pages 255–263. Springer, 2001.

[8] P. Lorenzen. Logik und agon. In *Arti del XII Congresso Internationale de Filosofia*, pages 187–194, 1958.

[9] H. Rückert. *Dialogues as a Dynamic Framework for Logic*. PhD thesis, Universiteit Leiden, 2007.

[10] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, 2nd edition, 2000.

[11] UnCommon Web — Dynamic HTML Generation. `http://common-lisp.net/project/ucw/`.