# User-Computer Persuasion Dialogue for Grounded Semantics[1]

Martin Caminada [a] Mikołaj Podlaszewski [a]

[a] *Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg*

We present an implementation of the recently developed persuasion dialogue game for grounded semantics [2]. The idea is to apply Mackenzie-style dialogue [3, 4] to convince the user that an argument is or is not in the grounded extension. Our approach is based on the concept of a complete labelling [1], which is essentially a function that assigns each argument a label that is either `in`, `out` or `undec`, such that for each argument it holds that

- the argument is labelled `in` iff all its attackers are labelled `out`
- the argument is labelled `out` iff it has at least one attacker that is labelled `in`

Standard argumentation theory states that an argument is in the grounded extension iff it is labelled `in` by each and every complete labelling [1]. Therefore, in order to convince someone that an argument is in the grounded extension, it suffices to show that the fact that it has to be labelled `in` follows from the definition of a complete labelling. For this, we apply the concept of Mackenzie-style dialogue [3, 4]. Our theory differs from the Standard Grounded Game [5] in that (1) we apply Mackenzie-style dialogue moves, like `claim`, `why` `because` and `concede`, (2) when an argument is labelled `in`, we show that *all* its attackers are labelled `out` whereas in the Standard Grounded Game this is shown for only one of the attackers (at least in a single game or line of arguments), (3) we rely on the concept of a commitment store for determining the possible moves and ensuring correctness and completeness w.r.t. grounded semantics, (4) we do not apply the notion of a discussion tree, which after all is alien to Mackenzie-style dialogue, and (5) the presence of a winning strategy is not required to establish membership of the grounded extension; instead a single game won by the proponent against a maximally skeptical opponent is sufficient.

Our implementation uses a command-line interface, and is written in Python. The argumentation framework can either be loaded from a text file or entered manually. At the highest level, the user has eight commands at his disposal: `question`, `claim`, `load`, `save`, `af_cat`, `af_define`, and `quit`. With `question` the user asks the system about the status of a particular argument (say $A$). The system then responds either with `claim in`$(A)$, meaning that $A$ has to be labelled `in` by every complete labelling (hence, $A$ is in the grounded extension), with `claim out`$(A)$, meaning that $A$ has to be labelled `out` by every complete extension (hence, $A$ is attacked by the grounded extension) or with `no commitment` $A$, meaning that neither is the case. In the first two cases, the associated `claim` move is the start of a grounded dialogue as described in [2], which the user could choose to bypass by immediately conceding the main claim. When the user does a `claim` command, the system responds either by conceding (if it holds the claim that a particular argument has to be labelled `in` or `out` to be correct) or by holding a persuasion dialogue (if the system holds the claim to be incorrect). Although in the latter case, the discussion will in the end always be won by the system (since the ability to win the grounded persuasion game for a particular argument coincides with its membership of the grounded extension [2]) the discussion might still lead the user to valuable insight about why his initial position was wrong. With the `load`, `save`, `af_cat` and `af_define` commands one respectively loads, saves, displays or manually defines an argumentation framework. The dialogue game follows the rules described in [2], with the exception that parties can terminate the dialogue at any point by conceding or withdrawing the main claim.

Assume the argumentation framework of Figure 1. The interaction between the system and the user may look as follows.

[1] Supported by the National Research Fund, Luxembourg (LAAMI and LAAMIcomp projects)

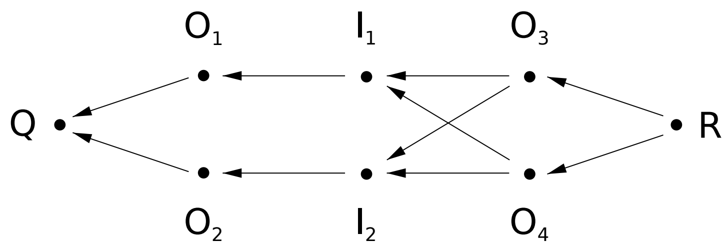


Figure 1: An Argumentation framework with multiple lines

| Moves | Commitment | | | |
|---|---|---|---|---|
| | Proponent | | Opponent | |
| | `in` | `out` | `in` | `out` |
| U: `question` $Q$ | - | - | - | - |
| C: `claim` $\texttt{in}(Q)$ | $Q$ | - | - | - |
| U: `why` $\texttt{in}(Q)$ | $Q$ | - | - | - |
| C: `because` $\texttt{out}(O_1, O_2)$ | $Q$ | $O_1, O_2$ | - | - |
| U: `why` $\texttt{out}(O_1)$ | $Q$ | $O_1, O_2$ | - | - |
| C: `because` $\texttt{in}(I_1)$ | $Q, I_1$ | $O_1, O_2$ | - | - |
| U: `why` $\texttt{in}(I_1)$ | $Q, I_1$ | $O_1, O_2$ | - | - |
| C: `because` $\texttt{out}(O_3, O_4)$ | $Q, I_1$ | $O_1, O_2, O_3, O_4$ | - | - |
| U: `why` $\texttt{out}(O_3)$ | $Q, I_1$ | $O_1, O_2, O_3, O_4$ | - | - |
| C: `because` $\texttt{in}(R)$ | $Q, I_1, R$ | $O_1, O_2, O_3, O_4$ | - | - |
| U: `concede` $\texttt{in}(R)$, $\texttt{out}(O_3, O_4)$, $\texttt{in}(I_1)$, $\texttt{out}(O_1)$ | $Q, I_1, R$ | $O_1, O_2, O_3, O_4$ | $R, I_1$ | $O_3, O_4, O_1$ |
| U: `why` $\texttt{out}(O_2)$ | $Q, I_1, R$ | $O_1, O_2, O_3, O_4$ | $R, I_1$ | $O_3, O_4, O_1$ |
| C: `because` $\texttt{in}(I_2)$ | $Q, I_1, R, I_2$ | $O_1, O_2, O_3, O_4$ | $R, I_1$ | $O_3, O_4, O_1$ |
| C: `concede` $\texttt{in}(I_2), \texttt{out}(O_2), \texttt{in}(Q)$ | $Q, I_1, R, I_2$ | $O_1, O_2, O_3, O_4$ | $R, I_1, I_2, Q$ | $O_3, O_4, O_1, O_2$ |

Note that the argument $O_4$ is directly conceded (without playing a `why` move), because the argument $R$ was given as an answer to `why` $\texttt{out}(O_3)$. In general one can notice that each argument appears in dialogue at most three times - once in a `because` (`claim`) move, at most once in a `why` move and once in a `concede` move - hence the length of the dialogue is linear in number of arguments. In contrast, applying the Standard Grounded Game [5] would require investigation of four lines: $Q$-$O_1$-$I_1$-$O_3$-$R$, $Q$-$O_1$-$I_1$-$O_4$-$R$, $Q$-$O_2$-$I_2$-$O_3$-$R$, $Q$-$O_2$-$I_2$-$O_4$-$R$. Extending the example by duplicating four arguments $I_1, I_2, O_3, O_4$ will double this number and in general case the number of lines of the Standard Grounded Game is exponential w.r.t the number of arguments.

The source code (GPL) and other necessary files can be downloaded at our project page [2]. Our plan is to keep developing it and integrate it with ArguLab [6]. Furthermore, we plan to implement a similar dialogue game for credulous preferred semantics.

# References

[1] M.W.A. Caminada and D.M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009. Special issue: new ideas in argumentation theory.

[2] M.W.A. Caminada and M. Podlaszewski. A dialogue game for grounded semantics. Submitted to COMMA 2012.

[3] J. D. Mackenzie. Question-begging in non-cumulative systems. *Journal of Philosophical Logic*, 8:117–133, 1979.

[4] J. D. Mackenzie. Four dialogue systems. *Studia Logica*, 51:567–583, 1990.

[5] S. Modgil and M.W.A. Caminada. Proof theories and algorithms for abstract argumentation frameworks. In I. Rahwan and G.R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 105–129. Springer, 2009.

[6] M. Podlaszewski, Y. Wu, and M. Caminada. An implementation of basic argumentation components. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1307–1308. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[2] `http://code.google.com/p/pyafl/downloads/list`