Playing extensive form negotiation games:
A tool-based analysis (Abstract)

Sujata Ghosh[1], Sumit Sourabh[2] and Rineke Verbrugge[1]

[1] Department of Artificial Intelligence
University of Groningen
PO Box 407, 9700 AK Groningen
The Netherlands.
`sujata,rineke@ai.rug.nl`
[2] Department of Mathematics and Statistics
Indian Institute of Technology
Kanpur 208 016, India.
`sumit.sourabh@gmail.com`

Introduction. This paper reports the development of a simple tool, NEGEXT, written in the platform-independent Java language. NEGEXT has been constructed to aid real people doing actual negotiations, when the ways to negotiate are simply too many to be computed by a normal human mind. This toolkit will also help in planning one's strategic moves in negotiation situations when the opponents' possible moves can be approximated. Even though some visualization tools for extensive form game trees already exist, we believe we are the first to make a tree-based negotiation toolkit that incorporates the possibility of representing learning from game to game, by sequential and parallel composition (cf. [1]). Moreover, the toolkit has a model-checking component which computes whether and how an individual or a specific coalition can achieve a given objective.

NEGEXT toolkit We have developed the NEGEXT toolkit[3] to represent and aid in the strategic interactions that happen during negotiations. Many negotiation situations can be aptly represented as extensive form games and their sequential and parallel combinations.

The software is represented by a Game tree simulator (Figure 1), written in Java version 1.6.0_22. It uses an applet for displaying the graphical user interface and can run on any system having a Java Runtime Environment (JRE) installed or Java-enabled web browser. The users are provided with an option to draw game trees using different input modes. The trees thereby generated are rendered using Graphviz for the final output. NEGEXT provides an option to combine the trees sequentially and in parallel. The main feature of this software is to check for the existence of a strategy for achieving a formula for an individual or a set of players, as well as displaying that strategy (or all such strategies), in case one exists (Figure 1).

---

[3] available at `http://www.ai.rug.nl/~sujata/negext.html`
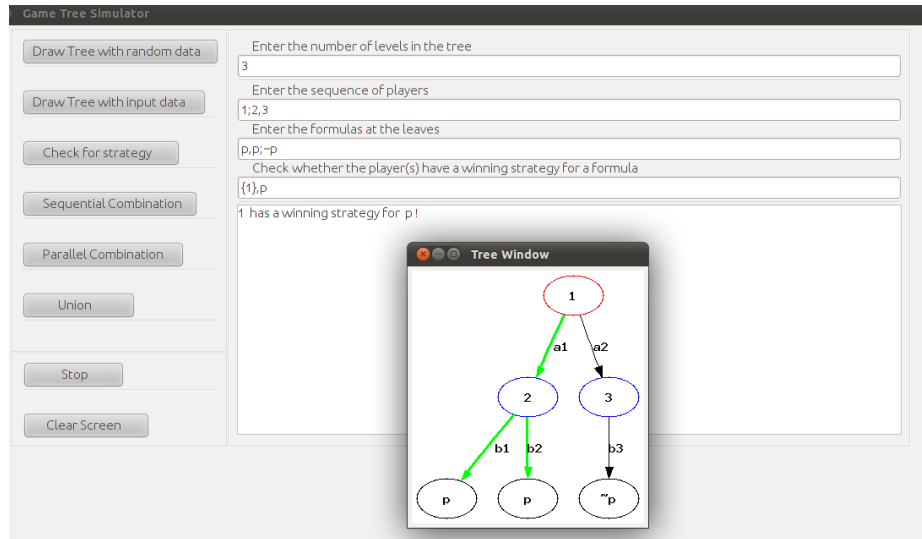
Fig. 1. Strategy checking in the graphical user interface

Discussions. This toolkit can be used to represent negotiations which span over a finite time and consider the actions of the negotiators one after another in response to each other. The NEGEXT toolkit can help users to learn how to respond in order to achieve what they want, also in situations where it is not easy to compute the best response. We can consider simple game trees, build up complex game structures from those trees by sequential and parallel combination, and then compute group and individual strategies.

The current version of NEGEXT is restricted to perfect information situations. However, in many real-life negotiations, the information dilemma looms large: Which aspects to make common knowledge and which aspects to keep secret, or to divulge only to a select subset of co-players? It would be fascinating to extend NEGEXT so that also such aspects of incomplete, imperfect and asymmetric information can be incorporated in its tree representations and its strategic advice, but that is still in the future.

References

1. S. Ghosh, R. Ramanujam, and S. E. Simon. Playing extensive form games in parallel. In J. Dix and others, editors, *CLIMA XI*, volume 6245 of *LNCS*, pages 153–170, Berlin, 2010. Springer.