Available online at www.sciencedirect.com
SCIENCE DIRECT®
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 141 (2005) 135–162
www.elsevier.com/locate/entcs

Interaction in Normative Multi-Agent Systems

Guido Boella[1]

Dipartimento di Informatica
Università di Torino
Torino, Italy

Joris Hulstijn[2]

Faculty of Economics and Business Administration
Vrije Universiteit
Amsterdam, The Netherlands

Leendert van der Torre[3]

CWI Amsterdam
and Delft University of Technology
The Netherlands

Abstract

The central research question of this paper is how notions developed in interactive computing such as abstract behavior types, the coordination language Reo, and Boolean circuits with registers, can be used to extend logical input/output nets, or lions for short. Lions are based on input/output logic, a deontic logic which is not used as a (non-classical) inference engine deriving output from input, but as a secretarial assistant for logically assisted transformations from input to output. We consider two extensions of input/output logics and lions. First, we consider input/output logics defined on infinite sequences (or streams) of inputs and outputs. Secondly, we consider lions with AND and register gates, formalizing the behavior of channels and connectors. We discuss also the role of interactive computing in normative multi-agent systems motivating the development of lions.

Keywords: Normative systems, multi-agent systems, deontic logic, input/output logic, coordination, communication.

1571-0661/$ – see front matter
doi:10.1016/j.entcs.2005.05.020

1 Introduction

According to many in computer science, the interaction paradigm provides a new conceptualization of computational phenomena that emphasize interaction rather than algorithms: concurrent, distributed, reactive, embedded, component-oriented, agent-oriented and service-oriented systems all exploit interaction as a fundamental paradigm [34,5].

In this paper we consider Makinson and van der Torre's logical input/output nets [25], or lions for short, as a model for interactive computation. Lions are structured assemblies of input/output operations and extend Makinson and van der Torre's input/output logics [22,23]. They are graphs, with the nodes labelled by pairs (G, out) where G is a normative code and out is an input/output operation. The edges of the graph represent channels, which indicate which nodes have access to which other nodes and provide passage for the transmission of local outputs as local inputs. The graph is further equipped with an entry point and an exit point, for global input and output.

We consider also two extensions of lions in this paper, both inspired by the work of Arbab and colleagues on abstract behavior types [4], the coordination language Reo [3], and Boolean circuits with registers [28].

Streams. Instead of considering one input at a time, we consider input/output logics on infinite sequences (or streams) of inputs and outputs.

Connectors. We consider AND and register gates to compose channels into connectors (or circuits).

Finally we discuss the role of interaction in normative multi-agent systems, and how it is used to motivate the further development of lions. Our investigations reveal a huge number of possible further extensions of lions, which raises the question which extensions should be studied next. Input/output logic originates from deontic logic, a branch of philosophical logic that studies logical relations among obligations, permissions and prohibitions, and which has been used to model legal and moral systems, as well as problems in computer science that involve constraints that can be violated [35]. Whereas deontic logic has been very helpful in the development of input/output logics, it does not seem very helpful to guide the development of lions. A motivation of lions comes from agent architectures and normative multi-agent systems, i.e., "sets of agents (human or artificial) whose interactions can fruitfully be regarded as norm-governed; the norms prescribe how the agents ideally should and should

[1] Email: guido@di.unito.it
[2] Email: jhulstijn@feweb.vu.nl
[3] Email: torre@cwi.nl