SOMEWHERE2 – A Robust Package for Collaborative Decentralized Consequence-Finding

Philippe Chatalic and Andre de Amorim Fonseca

Abstract. This paper presents SOMEWHERE2, a new framework that may be used for setting up peer-to-peer inference systems and for solving consequence finding problems in a completely decentralized way. It is a complete redesign and reengineering of an earlier platform. The new architecture has gained in genericity, modularity and robustness. It is much easier to extend and/or to reuse as a building block for advanced distributed applications, such as Peer Data Management Systems.

1 Introduction

The consequence finding problem [2, 10] amounts to finding formulas that are consequences of a logical theory. Many applications involve reasoning tasks that aim at discovering such consequences, not explicit in the original theory. Often, not all consequences are sought, but only a subset of those, satisfying some syntactical property, called a production field [12]. Consequence finding is more complex than the proof finding problem, for which a user simply wants to verify whether a formula is entailed or not by a theory. It has proved to be useful for wide range of problems involving diagnosis, abductive reasoning, hypothetical and non-monotonic reasoning, query rewriting as well as knowledge compilation (see [10] for a survey).

There are several reasons to consider this problem in a distributed setting. For large theories, the problem may rapidly become out of scope for a single computing unit. Exploiting structural properties of the original theory in order to decompose it into subparts is one possible approach. It has been explored in the context of theorem proving by [2] and recently extended to the case of consequence finding in [3].

Philippe Chatalic
L.R.I. - Bat 650, Université Paris-Sud, Orsay, France
e-mail: chatalic@lri.fr

Andre de Amorim Fonseca
L.R.I. , Inria Saclay Ille-de-France, Orsay, France
e-mail: andre.amorimfonseca@gmail.fr
But the need for a distributed approach becomes essential when the knowledge is intrinsically scattered at different places. This is the case in some multi agent architectures, where each agent is not necessarily willing (e.g. for some privacy reasons) to share all of its knowledge, but has to collaborate with others in order to achieve its goals. Similarly, semantic data management systems exploit the content of multiple sources of information, each of them being described using its own ontology. Query answering over such networked data sources requires reasoning over distributed ontologies. It generally proceeds in two steps, the first of which is a query rewriting step (that can be reformulated as a consequence finding problem), where the original query is rewritten in terms of the languages of the different relevant ontologies. Obtained rewritings are then evaluated on the appropriate sources.

Given the ever growing number of information sources available over the web, peer-to-peer (P2P) architectures look particularly promising for that purpose. The absence of any centralized control or hierarchical organization and the fact that each peer plays the same role gives much flexibility to accommodate to the dynamic nature of such networks. This also contributes to the scalability and the robustness of such approaches. Such principles are at the core of Peer Data Management Systems (PDMS) such as Edutella [11], Piazza [5] or Somewhere [1].

Somewhere is a framework based on a decentralized propositional P2P inference system (P2PIS) that can be used as a corner stone for designing elaborated PDMS. Its scalability on fairly large networks of peers has been very encouraging. It is however rather a proof of concept than a rock solid piece of code. Unstable, it missed essential features, e.g. the ability to cope with the dynamicity of the network. Moreover, costs for its maintenance and attempts to add new features turned out to be extremely high. At some time, the best solution has appeared to start a complete reengineering, in order to improve both its design, robustness and extensibility. The main contribution of this paper is to present the core architecture of this new system.

2 Consequence Finding in P2P Inference Systems

Somewhere is based on a decentralized consequence finder that consider P2PIS \( \mathcal{P} = \{P_i\}_{i=1..n} \) such as the one of Fig. 1, where each peer has its own vocabulary \( V_i \) (a set of propositional variables) and a local clausal theory \( P_i = O_i \cup M_i \), \( O_i \) denotes the set of local clauses, that are made exclusively of literals over \( V_i \) (here symbols indexed by \( i \)), while \( M_i \) denotes mapping clauses, involving the vocabulary of at least two different peers. Intuitively, local clauses describe the very own knowledge of the peer \( P_i \) while mappings state logical constraints between different peer theories. Variables appearing in several peers are said to be shared (edges labels on fig. 1). They characterize possible interactions between peers and implicitly define an acquaintance graph. We assume each peer to be aware of its acquaintances and denote by \( \text{ACQ}(l, P_i) \) the set of peers with which \( P_i \) shares the variable of a literal \( l \). The global theory \( P = \bigcup_{i=1..n} P_i \) is a set of clauses over the vocabulary \( V = \bigcup_{i=1..n} V_i \).

For such networks, we consider the classical semantics of propositional logic. We use \( \models \) to denote the classical consequence relation. A clause \( c \) is an implicate of