J Autom Reasoning (2014) 53:245–269
DOI 10.1007/s10817-014-9305-1

# HermiT: An OWL 2 Reasoner

**Birte Glimm · Ian Horrocks · Boris Motik ·
Giorgos Stoilos · Zhe Wang**

Received: 12 July 2013 / Accepted: 27 March 2014 / Published online: 23 May 2014


**Abstract** This system description paper introduces the OWL 2 reasoner HermiT. The reasoner is fully compliant with the OWL 2 Direct Semantics as standardised by the World Wide Web Consortium (W3C). HermiT is based on the hypertableau calculus, and it supports a wide range of standard and novel optimisations that improve the performance of reasoning on real-world ontologies. Apart from the standard OWL 2 reasoning task of entailment checking, HermiT supports several specialised reasoning services such as class and property classification, as well as a range of features outside the OWL 2 standard such as DL-safe rules, SPARQL queries, and description graphs. We discuss the system's architecture, and we present an overview of the techniques used to support the mentioned reasoning tasks. We further compare the performance of reasoning in HermiT with that of FaCT++ and Pellet—two other popular and widely used OWL 2 reasoners.

B. Glimm
Institute of Artificial Intelligence, University of Ulm, 89069 Ulm, DE, Germany
e-mail: birte.glimm@uni-ulm.de

I. Horrocks · B. Motik (✉)
Department of Computer Science, University of Oxford,
Wolfson Building, Parks Road, Oxford, OX1 3QD, UK
e-mail: boris.motik@cs.ox.ac.uk

I. Horrocks
e-mail: ian.horrocks@cs.ox.ac.uk

G. Stoilos
School of Electrical & Computer Engineering, National Technical University of Athens,
Iroon Polytexneiou 9, 15780 Zografou, Greece
e-mail: gstoil@image.ntua.gr

Z. Wang
School of Information & Communication Technology, Griffith University,
Brisbane, QLD 4111, Australia
e-mail: zhe.wang@griffith.edu.au

**Keywords** Ontologies · OWL · Class classification · Property classification · Optimisations

## 1 Introduction

In this system description paper we describe the main features of the HermiT ontology reasoner. HermiT supports all features of the OWL 2 ontology language [4], including all OWL 2 datatypes [26], and it correctly performs both object and data property classification—reasoning tasks that are, to the best of our knowledge, not fully supported by any other OWL reasoner. In addition to these standard reasoning tasks, HermiT also supports SPARQL query answering, and it uses a range of optimisations [21] to ensure efficient processing of real-world ontologies. Furthermore, HermiT supports several features that go beyond existing standards, such as DL-safe SWRL rules [15, 29] and description graphs [24]—an extension of OWL 2 that allows for a faithful modelling of arbitrarily connected structures.

A key novel idea in HermiT is the hypertableau calculus [30], which allows the reasoner to avoid some of the nondeterministic behaviour exhibited by the tableau calculus used in Pellet [36] and FaCT++ [39]—two other popular and widely used OWL reasoners. In order to further improve the performance of the calculus, HermiT employs a wide range of standard and novel optimisation techniques, including anywhere blocking [30], blocking signature caching [30], individual reuse [25], and core blocking [7]. HermiT also implements a novel classification algorithm [8] that greatly reduces the number of consistency tests needed to compute the class and property hierarchies.

We have compared HermiT's performance with Pellet [36] and FaCT++ [39] on a set of standard ontologies. Unlike many earlier evaluations, all the ontologies used in our tests are directly accessible via immutable URIs, so our tests are fully repeatable and their results are unambiguous. Our results show that, although HermiT did not outperform the other reasoners on all ontologies, it seemed more robust as it managed to process more 'hard' ontologies.

The rest of this paper is organised as follows: in Section 2 we introduce HermiT's system architecture; in Section 3 we present an overview of the hypertableau calculus; in Section 4 we discuss several optimisations of the core calculus; in Section 5 we outline the features that go beyond OWL 2 and discuss their support in the reasoner; and in Section 6 we evaluate HermiT's performance on a wide range of ontologies and compare it to the performance of Pellet and FaCT++.

We assume the reader to be familiar with OWL, description logics [12, 22], and the correspondence between the OWL and the description logic syntax [14]. We take an OWL ontology $\mathcal{O}$ to consist of a *TBox* $\mathcal{T}$ and an *ABox* $\mathcal{A}$, where the former specifies the schema (i.e., the axioms that describe the structure of the domain being modelled) and the latter contains the data (i.e., the assertions describing the objects in a domain of discourse). For brevity we will mainly use the standard description logic syntax; however, as HermiT is an OWL reasoner, we will talk about *classes* and *properties*, which are commonly called *concepts* and *roles* in the description logic community.

## 2 System Architecture

HermiT consists of several components that together implement a sound and complete OWL reasoning system. Figure 1 shows the most important components (e.g.,