Automated Reasoning
and Ontology

Jeff Z. Pan, Jianfeng Du

Synonyms

Logical reasoning, inference, semantic computing, approximate reasoning

Definitions

Reasoning is the process of deriving conclusions in a logical way. Automatic reasoning is concerned with the construction of computing systems that automate this process over some knowledge bases.

Automated Reasoning is often considered as a sub-field of Artificial Intelligence. It is also studied in the fields of theoretical computer science and even philosophy.

The development of formal logic (Frege [1884]) played a big role in the field of automated reasoning, which itself led to the development of artificial intelligence.

Historically, automated reasoning is largely related to theorem proving, general problem solvers and expert systems (cf. the section of ‘A Bit of History’). In the context of big data processing, automated reasoning is more relevant to modern knowledge representation languages, such as the W3C standard Web Ontology Language OWL (https://www.w3.org/TR/owl2-overview/), in which a knowledge base consists of a schema component (TBox) and a data component (ABox).

From the application perspective, perhaps the most well known modern knowledge representation mechanism is Knowledge Graph (Pan et al (2016b, 2017)). In 2012, Google popularised the term ‘Knowledge Graph’ by using it for improving its search engine. Knowledge graphs are then adopted by most leading search engines (such as Bing and Baidu) and many leading IT companies (such as IBM and Facebook). The basic idea of Knowledge Graph is based on the knowledge representation formalism called Semantic Networks. There is a modern W3C standard for semantic networks called RDF (Resource Description Framework, https://www.w3.org/TR/rdf11-concepts/). Thus RDF/OWL graphs can be seen as exchangeable knowledge graphs in the big data era.

Jianfeng Du
Guangdong University of Foreign Studies, e-mail: jfdu@gdufs.edu.cn
While this entry will be mainly about automated reasoning techniques in the big data era, their classifications, key contributions, typical systems, as well as their applications, it starts with a brief introduction of the history.

A Bit of History

Many consider the Cornell Summer Meeting of 1957, which brought together many logicians and computer scientists, as the origin of automated reasoning.

The first automated reasoning systems were theorem provers, systems that represent axioms and statements in First Order Logic and then use rules of logic, such as modus ponens, to infer new statements. The first system of this kind is the implementation of Presburger’s decision procedure (which proved that the sum of two even numbers is even) by Davis (1957).

Another early type of automated reasoning system were general problem solvers, which attempt to provide a generic planning engine that could represent and solve structured problems, by decomposing problems into smaller more manageable sub-problems, solving each sub-problem and assembling the partial answers into one final answer. The first system of this kind is Logic Theorist from Newell et al (1957).

The first practical applications of automated reasoning were expert systems, which focused on much more well-defined domains than general problem solving, such as medical diagnosis or analysing faults in an aircraft, and on more limited implementations of First Order Logic, such as modus-ponens implemented via IF-THEN rules. One of the forerunners of these systems is MYCIN by Shortliffe (1974).

Since 1980s, there have been prosperous studies of practical subsets of First Order Logics as ontology languages, such as Description Logics (Baader et al (2003)) and Answer Set Programming (Lisitsa (2002)), as well as the standardisation of ontology language OWL (version 1 in 2004 and version 2 in 2009). The wide adoption of Ontology and Knowledge Graph (Pan et al (2016b, 2017)), including by Google and many other leading IT companies, confirms the status of ontology language in big data era.

In the rest of the entry, we will focus on automated reasoning with ontology languages.

Classification

There can be different ways of classifying research problems related to automated ontology reasoning.

From the purpose point of view, automatic ontology reasoning can be classified into (1) deductive ontology reasoning (Levesque and Brachman (1987)), which draws conclusions from given premises, (2) abductive ontology reasoning (Colucci et al (2003)), which finds explanations for observations that are not consequences of given premises, as well as (3) inductive ontology reasoning (Lisi and Malerba (2003)), which con-
cludes that all instances of a class has a certain property if some instances of the class has the property.

From the direction point of view, automatic ontology reasoning can be classified into (1) forward reasoning (Baader et al. 2005), in which the inference starts with the premises, moves forward and ends with the conclusions, (2) backward reasoning (Grosof et al. 2003), in which the inference starts with the conclusions, moves backward and ends with the premises, as well as (3) bi-directional reasoning (MacGregor 1991) in which the inference starts with both the premises and the conclusions, moves forward and backward simultaneously or interactively, until the intermediate conclusions obtained by forward steps include all intermediate premises required by backward steps.

From the monotonicity point of view, automatic ontology reasoning can be classified into (1) monotonic ontology reasoning in which no existing conclusions will be dropped when new premises are added, as well as (2) nonmonotonic ontology reasoning (Quantz and Suska 1994) in which some existing conclusions can be dropped when new premises are added.

From the scalability point of view, automatic ontology reasoning can be classified into (1) parallel ontology reasoning (Bergmann and Quantz 1995), in which reasoning algorithms can exploit multiple computation cores in a computation nodes and (2) distributed ontology reasoning (Borgida and Serafini 2003), L Serafini (2005), in which reasoning algorithms can exploit a cluster of computation nodes. Scalable ontology reasoning is also often related to strategies of modularisation (Suntisrivaraporn et al. 2008) and approximation (Pan and Thomas 2007).

From the mobility point of view, automated ontology reasoning can be classified into (1) reasoning with temporal ontologies (Artale and Franconi 1994), in which the target ontologies contain temporal constructors for class and property descriptions, and (2) stream ontology reasoning (Stuckenschmidt et al. 2010; Ren and Pan 2011), which, given some continuous updates of the ontology, requires updating reasoning results without naively re-computing all results.

From the certainty point of view, automatic reasoning can be classified into (1) ontology reasoning with certainty in which both premises and conclusions are certain and either true or false, as well as (2) uncertainty ontology reasoning (Koller et al. 1997) in which either premises or conclusions are uncertain and often have truth values between 0/-1 and 1. There are different kind of uncertainties within ontologies, such as probabilistic ontologies (Koller et al. 1997), fuzzy ontologies (Straccia 2001) and possibilistic ontologies (Qi et al. 2011).

Key Contributions

The highlight on contributions of automated ontology reasoning is the standardisation of the Web Ontology Language OWL.
The first version of OWL (or OWL 1) was standardised in 2004. It is based on the SHOIQ DL (Horrocks and Sattler (2005)). However, there are some limitations of OWL 1:

1. the datatype support is limited (Pan and Horrocks (2006));
2. the only sub-language, OWL-Lite, of OWL 1 is not tractable;
3. the semantics of OWL 1 and RDF are not fully compatible (Pan and Horrocks (2003)).

The second version of OWL (or OWL 2) was standardised in 2009. It is based on the SROIQ DL (Horrocks et al (2006)). On the one hand, OWL 2 has more expressive power, such as the stronger support of datatypes (Pan and Horrocks (2006); Motik and Horrocks (2008)) and rules (Krtzsch et al (2008)). On the other hand, OWL 2 has three tractable sub-languages, including OWL 2 EL (Baader et al (2005)), OWL 2 QL (Calvanese et al (2007)) and OWL 2 RL (Grosof et al (2003)).

This two-layer architecture of OWL 2 allows approximating OWL 2 ontologies to those in its tractable sub-languages, such as approximations towards OWL 2 QL (Pan and Thomas (2007), towards OWL 2 EL (Ren et al (2010)) and towards OWL 2 RL (Zhou et al (2013)), so as to exploit efficient and scalable reasoners of the sub-languages. The motivation is based on the fact that real-world knowledge and data are hardly perfect or completely digitalised.

**Typical Reasoning Systems**

Below are descriptions of some well-known OWL reasoners (in alphabetical order).

**CEL**

CEL (Baader et al (2006)) is a LISP-based reasoner for $\mathcal{EL}^+$ (Baader et al (2008)), which covers the core part of OWL 2 EL. CEL is the first reasoner for the description logic $\mathcal{EL}^+$, supporting as its main reasoning task the computation of the subsumption hierarchy induced by $\mathcal{EL}^+$ ontologies.

**ELK**

ELK (Kazakov et al (2012)) is an OWL 2 EL reasoner. At its core, ELK uses a highly optimised parallel algorithm (Kazakov et al (2011)). It supports stream reasoning in OWL 2 EL (Kazakov and Klinov (2013)).

**FaCT**

FaCT (Horrocks (1998)) is a reasoner for the description logic $\mathcal{SHIF}$ (OWL-Lite). It is the first modern reasoner that demonstrates the feasibility of using optimised algorithms for subsumption checking in realistic applications.

**FaCT++**

FaCT++ (Tsarkov and Horrocks (2006)) is a reasoner for (partially) OWL 2. It is the new generation
of the well-known FaCT reasoner is implemented using C++, with a different internal architecture and some new optimisations.

HermiT
HermiT (Glimm et al.(2014)) is a reasoner for OWL 2. It is the first publicly available OWL 2 reasoner based on a hypertableau calculus (Motik et al. (2009)), with a highly optimised algorithm for ontology classification (Glimm et al. (2010)). HermiT can handle DL Safe rules (Motik et al. (2005)) on top of OWL 2.

Konclude
Konclude (Steigmiller et al. (2014)) is a reasoner for OWL 2. It supports almost all datatypes in OWL 2. Konclude implements a highly optimised version of tableau calculus enhanced with tableau saturation (Steigmiller and Glimm (2015)). It supports parallel reasoning and nominal schemas (Steigmiller et al. (2014a)) and DL-safe rules.

Mastro
Mastro (Calvanese et al. (2011)) is an Ontology-Based Data Access (OBDA) management system for OWL 2 QL. It allows data to be managed external relational data management or data federation systems. It uses the Presto algorithm (Rosati and Almatelli (2010)) for query rewriting.

Ontop
Ontop (Calvanese et al. (2016)) is an Ontology-Based Data Access (OBDA) management system for RDF and OWL 2 QL, as well as SWRL with limited forms of recursions. It also supports efficient SPARQL-to-SQL mappings via R2RML (Rodriguez-Muro and Rezk (2015)). Ontop has some optimisations on query rewriting based on database dependencies (Rodriguez-Muro et al. (2013)).

Pellet
Pellet (Sirin et al. (2007)) is a reasoner for OWL 2. It also has dedicated support for OWL 2 EL. It incorporates optimisations for nominals, conjunctive query answering, and incremental reasoning.

Racer
Racer (Haarslev and Müller (2001)) is a reasoner for OWL 1. It has a highly optimised version of tableau calculus for the description logic $SHI Q(D)$ (Horrocks and Patel-Schneider (2003)).

RDFox
RDFox (Motik et al. (2014)) is a highly scalable in-memory RDF triple store that supports shared memory parallel datalog (Ceri et al. (1989)) reasoning. It supports stream reasoning (Motik et al. (2015b)) and
has optimisations for owl:sameAs (Motik et al. (2015a)).

**TrOWL**

TrOWL (Thomas et al. (2010)) is a highly optimised approximate reasoner (Pan et al. (2016a)) for OWL 2. TrOWL outperforms some sound and complete reasoners in the time-constrained ORE (Ontology Reasoner Evaluation) competitions designed for sound and complete ontology reasoners. TrOWL has stream reasoning capabilities for both OWL 2 and OWL 2 EL (Ren and Pan (2011); Ren et al. (2016)). It supports local closed world reasoning in NBox, or closed predicates (Lutz et al. (2013)).

**Applications**

Automated ontology reasoning has been widely used in Web applications, such as for content management (BBC), travel planning and booking (Skyscanner), web search (Google, Bing, Baidu).

It is also being applied in a growing number of vertical domains. One typical example is life science. For instance, OBO foundry includes more than 100 biological and biomedical ontologies. The SNOMED-CT (Clinical Terminology) ontology is widely used in healthcare systems of over 15 countries, including US, UK, Australia, Canada, Denmark and Spain. It is also used by major US providers, such as Kaiser Permanente. Other vertical domains include, but not limited to, agriculture, astronomy, oceanography, defence, education, energy management, geography and geoscience.

While ontologies are widely used as structured vocabularies, providing integrated and user-centric view of heterogeneous data sources in the big data era, benefits of using automated ontology reasoning include:

1. Reasoning support is critical for development and maintenance of ontologies, in particular on derivation of taxonomy from class definitions and descriptions.
2. Easy location of relevant terms within large structured vocabulary;
3. Query answers enhanced by exploiting schema and class hierarchy.

An example in the big data context is the use of ontology and automated ontology reasoning for data access in Statoil, where about 900 geologists and geophysicists use data from previous operations in nearby locations to develop stratigraphic models of unexplored areas, involving diverse schemata and TBs of relational data spread over 1000s of tables and multiple databases. Data analysis is the most important factor for drilling success. 30-70% of these geologists and geophysicists’ time is spent on data gathering. The use of ontologies and automated ontology reasoning enable better use of experts’ time, reducing turnaround for new queries significantly.
Outlook

Despite the current success of automated ontology reasoning, there are still some pressing challenges in the big data era, such as the following:

1. Declarative data analytics (Kaminski et al. (2017)) based on automated ontology reasoning;
2. Effective approaches of producing high quality (Ren et al. (2014); Konev et al. (2014)) ontologies and knowledge graphs;
3. Integration of automated ontology reasoning with data mining (Lecue and Pan (2015)) and machine learning (Chen et al. (2017)) approaches.

References

Ceri S, Gottlob G, Tancar L (1989) What you always wanted to know about Database (and never dared to ask). IEEE Transactions on Knowledge and Data Engineering 1:146–166
Frege G (1884) Die Grundlagen der Arithmetik. Breslau: Wilhelm Kobner
Lutz C, Seylan I, Wolter F (2013) Ontology-Based Data Access with Closed Predicates is Inherently Intractable (Sometimes). In: IJCAI2013, pp 1024–1030
Pan JZ, Ren Y, Zhao Y (2016a) Tractable approximate deduction for OWL. Artificial Intelligence 235:95–155


Pan JZ, Calvanese D, Eiter T, Horrocks I, Kifer M, Lin F, Zhao Y (2017) Reasoning Web: Logical Foundation of Knowledge Graph Construction and Querying Answering. Springer


