Extending ASP Based Reasoning to Expressive Constructive Description Logics

Loris Bozzato

Fondazione Bruno Kessler,
Trento, Italy
Email: bozzato@fbk.eu

***Abstract*—Constructive description logics provide different interpretations of description logics under constructive semantics. While several constructive semantics for description logics have been proposed, such logics have been mostly studied from the point of view of their formal properties. Practical applications of these logics have been proposed, but there has been no effort in applying them in general in knowledge representation and Semantic Web languages and tools (i.e., the distinctive applications of description logics). In our recent work, we started to address this aspect: we introduced a constructive semantics for the description logic $\mathcal{EL}$ and we established formal results that link this constructive interpretation to answer set semantics. On the base of these results, we presented a datalog translation reasoning on one aspect of such semantics (the generation of information terms of a knowledge base) and an implementation using Semantic Web ontologies and "off the shelf" tools. In this paper, we want to highlight this line of research and its possibilities and the challenges in extending this work to more expressive description logics.**

***Keywords*–Constructive description logics; Information terms semantics; Answer Set Programming.**

I. Introduction

Constructive description logics are interpretations of Description Logics (DLs) under different constructive semantics: the definition of such non-classical semantics for description logics is motivated by the interest in applying their formal properties to solve problems in representation of knowledge or reasoning. Starting from different constructive semantics, several constructive description logics have been proposed, e.g., [1]–[3]. Constructive description logics have been mostly studied from the point of view of their formal properties. However, there have been not many studies for their practical application to knowledge representation and Semantic Web languages and tools. On the other hand, different "real world" uses of constructive description logics in applications and systems have been proposed, for example in managing conflicts over legal ontologies [4], reasoning over incomplete data streams [5], and a framework for the composition of semantic services in heterogeneous domains [6] (based on [7]).

In our recent work [8], we proposed a direction for reducing the gap across the formal and practical aspects of constructive description logics. From the theoretical point of view, we introduced a minimal constructive description logic based on $\mathcal{EL}$ and we extended to its semantics formal results linking it to *Answer Set Programming (ASP)*. On the practical side, by taking advantage of such relation, we presented a datalog encoding managing one task over the constructive semantics (namely, the generation of valid "states" of a knowledge base) and we provided a prototype based on the standard *OWL EL (Web Ontology Language, EL profile)* and "off the shelf" tools.

In [8], we chose to concentrate on $\mathcal{EL}$, since it is one of the simplest description logics over which semantics enjoying constructive properties can be defined (cfr. *explicit definability* property [2]). Moreover, $\mathcal{EL}$ is recognized in applications as one of the reference languages for (low complexity) description logics (e.g., it is at the base of the OWL-EL profile).

We considered the task of the generation of valid *Information Terms (IT)* for a given knowledge base. Intuitively, in *information terms semantics* [2][9] the information terms are syntactical objects that provide a constructive justification for the classical truth of a formula. Information terms have been used to represent the *state* or *answer* of a formula: thus, generating information terms for a knowledge base corresponds to *validate* its contents by generating a set of its possible "snapshots" representing valid states. In particular, the solution in [8] follows the direction studied in [10], where the relation across information terms and answer set semantics has been first studied over propositional theories. In fact, we remark that [8] extends for the first time the study of these relations also in the context of constructive description logics.

Thus, from the formal point of view, the work in [8] can be seen as a starting point for the study of the connection across constructive semantics and answer set semantics on the very simple description logic $\mathcal{EL}$, positioning it as a base for extending the results over more expressive description logics by "pushing the envelope" towards more expressive languages.

In this position paper, we want to highlight this line of research and report the challenges in extending the current proposal to more expressive description logics. Indeed, while this would further move this approach closer to KR applications, the extension to expressive description logics constructs requires non-trivial formal work for the formulation of the relation between constructive semantics and ASP. Moreover, for the practical applicability of the approach, it is required to study efficient extensions of the ASP encoding.

In the following (in Section II), we summarize the most important aspects of the current work in this research direction. In Section III we discuss the challenges for future work.

II. Summary of Current Work

In [8], we introduced the logic $\mathcal{EL}$c, a constructive interpretation of $\mathcal{EL}$ under an information terms semantics. $\mathcal{EL}$c is a restriction to the syntax of $\mathcal{EL}$ of the basic constructive description logic $\mathcal{BCDL}$ [2][9]. With respect to the language, in order to simplify the definition of the semantics, in [8] we use a restricted form of subsumption $G \sqsubseteq A$ (denoted $\forall_G A$) where G is a *generator* (i.e., a fixed set of individuals). Moreover, the $\top$ operator is also limited to a fixed set $\mathcal{N}$ of objects. In the IT semantics, an information term η for a formula K is a syntactical object that provides a constructive *justification* for the truth of K in a classical model $\mathcal{M}$. For example, in a knowledge base of food and wines, the validity of a formula $\exists hasColor.Color(barolo)$ in a classical model $\mathcal{M}$ can be explained by an information term $(red, \mathtt{tt})$, where red is a valid role filler such that $hasColor(barolo, red)$ holds in $\mathcal{M}$ and $\mathtt{tt}$ is a (constant) information term inductively justifying the atomic formula $Color(red)$. This semantics is particularly interesting since, as studied in [2], the resulting *realizability relation* $\mathcal{M} \rhd \langle\eta\rangle\, K$ provides a constructive consequence that can be related to the *proofs-as-programs* paradigm.

Following the direction of [10], in our work we then provide formal results establishing a relation between answer sets for formulas in $\mathcal{EL}$ and for their information terms. In particular, we can show that (minimal) information terms for a set Γ of $\mathcal{EL}$ formulas can be obtained by computing the answer sets of input formulas in Γ. The relation can be established by recognizing that answer sets for a formula K correspond to $\mathtt{ans}(\langle\eta\rangle K)$, the “answers” of the *piece of information* $\langle\eta\rangle K$, composed by the minimal set of facts needed to justify $\langle\eta\rangle K$.

Using this correspondence, in [8] we provide two different datalog rewritings for the generation of the sets of information terms of an input $\mathcal{EL}$ knowledge base. The first is limited to the case in which the set of role assertions is “complete”, while the second provides a way of generating existential fillers. The encodings have been implemented in Asp-it [11], a prototype for an information terms generator for OWL EL ontologies: Asp-it applies the presented rewritings to the input ontology by interacting with the DLV solver and the resulting information terms are returned as annotations to the OWL axioms.

III. Discussion

For continuing this line of study, we can identify several parallel challenges that need to be considered:

Extension to unrestricted subsumption and $\top$. The first possible extension stands in the adoption in $\mathcal{EL}$c of all standard elements of the $\mathcal{EL}$ language, which have been restricted to facilitate the definition of the constructive semantics. As discussed in [2], the limited form of subsumption $\forall_G A$ in $\mathcal{EL}$c can be extended to consider general inclusions in the constructive semantics. However, it still need to be formally verified if the relation with ASP semantics can be extended to such general definition of subsumption. Similarly, allowing an unrestricted use of the $\top$ operator is non-trivial, due to the need to identify individuals in the definition of the IT semantics.

Semantic extension to expressive DL operators. A first step towards extending the approach to more expressive DLs is to introduce negation and falsum. Following [10], negative information can be represented similarly to default negation in logic programming: negative formulas are used as constraints and answer sets are formulated over a suitable positive *reduction* of the input formulas. This leads to a partition of answers for a piece of information $\mathtt{ans}(\langle\eta\rangle K)$ in the *positive* and *negative answers*, corresponding to positive and negative literals in the set: if the former correspond to “answers” of the piece of information, the latter represent the “constraints” that such positive answers must respect. Thus, the extension to $\mathcal{ALC}$ (corresponding to the $\mathcal{BCDL}$ logic [2]) is already challenging in establishing the correspondence results across answers sets of formulas and answers of pieces of information by considering the negative constraints. Following the approach of [10], we need to provide a definition of reduction to positive formulas in presence of negation and, on the other side, extend this notion to answers of pieces of information. Moreover, extending our semantics to expressive DL operators (e.g., to all operators of OWL, i.e., to $\mathcal{SROIQ}$) poses the question of providing a faithful IT interpretation for such operators.

ASP translation extension to expressive DLs. Clearly, the extension of the correspondence results to more expressive DLs can provide the base for extending the proposed ASP encoding. On the other hand, as already noted in [8], further study is already needed in the $\mathcal{EL}$ encoding to limit “combinatorial explosion” of the number of the models due to the generation of all admitted combinations of fillers for existential formulas. A possible direction in this regard is to study connections with existential extensions of datalog adding further constraints to limit the generation only to meaningful models.

Further reasoning tasks: manipulation of ITs. While the generation of information terms is a basic task for IT semantics, it only represents a first step towards the use of constructive description logics in practical applications. One fundamental direction (which is also related to the interesting relation of such semantics to the *proofs-as-programs* paradigm) would be to develop and integrate procedures that are able to manipulate the computed information terms. In this regard, a possibility is to study the applicability of this work in conjunction to the Semantic Web service composition calculus based on $\mathcal{BCDL}$ presented in [7].

Implementation and application to real use cases. Finally, another direction stands in the implementation of reasoning tasks and datalog encodings in available tools and languages in order to provide new tools for reasoning over constructive DLs. In this regard, it will be interesting to identify real use cases where the reasoning tasks can be applied and test the procedures over KBs based on well-known ontologies.

References

[1] L. Bozzato, M. Ferrari, C. Fiorentini, and G. Fiorino, “A decidable constructive description logic.” in JELIA 2010, ser. LNCS, vol. 6341. Springer, 2010, pp. 51–63.

[2] M. Ferrari, C. Fiorentini, and G. Fiorino, “$\mathcal{BCDL}$: basic constructive description logic,” JAR, vol. 44, no. 4, 2010, pp. 371–399.

[3] M. Mendler and S. Scheele, “Towards Constructive DL for Abstraction and Refinement,” JAR, vol. 44, no. 3, 2010, pp. 207–243.

[4] E. H. Haeusler, V. de Paiva, and A. Rademaker, “Intuitionistic description logic and legal reasoning,” in DEXA 2011 Workshops. IEEE Computer Society, 2011, pp. 345–349.

[5] M. Mendler and S. Scheele, “Towards a type system for semantic streams,” in SR2009 - Stream Reasoning Workshop (ESWC 2009), ser. CEUR-WP, vol. 466. CEUR-WS.org, 2009.

[6] M. Hilia, A. Chibani, K. Djouani, and Y. Amirat, “Semantic service composition framework for multidomain ubiquitous computing applications,” in ICSOC 2012, ser. LNCS, vol. 7636. Springer, 2012, pp. 450–467.

[7] L. Bozzato and M. Ferrari, “Composition of semantic web services in a constructive description logic.” in RR2010, ser. LNCS, vol. 6333. Springer, 2010, pp. 223–226.

[8] L. Bozzato, “ASP based generation of information terms for constructive $\mathcal{EL}$,” Fundam. Inform., vol. 161, no. 1-2, 2018, pp. 29–51.

[9] L. Bozzato, M. Ferrari, C. Fiorentini, and G. Fiorino, “A constructive semantics for $\mathcal{ALC}$.” in DL2007, ser. CEUR-WP, vol. 250. CEUR-WS.org, 2007, pp. 219–226.

[10] C. Fiorentini and M. Ornaghi, “Answer set semantics vs. information term semantics,” in ASP2007: Answer Set Programming, Advances in Theory and Implementation, 2007.

[11] “Asp-it: ASP based information terms generator,” 2016, URL: https://github.com/dkmfbk/asp-it [accessed: 2020-01-31].