Specifying the disjoint nature of object properties in DL

Cartik R. Kothari and David J. Russomanno
Department of Electrical and Computer Engineering
The University of Memphis, Memphis, TN 38152 USA
rkothari@memphis.edu, d-russomanno@memphis.edu

Abstract
This paper proposes constructs that can be used to declaratively specify the disjoint nature of object properties or roles. These constructs may be a useful extension to the Description Logic system that is the basis of OWL.

1 Introduction

The ability to specify disjoint relations has several applications in database and knowledge based systems. This paper introduces a set of syntactic constructs that can be used to specify the disjoint nature of roles in Description Logic (DL) [1] systems. Wessel [2] presents another study upon DL systems that specify the disjointness of roles; however, it did not specifically investigate the disjoint nature of roles. Instead, role disjointness was used as a starting premise to investigate the composition of roles in ALC_{RA} DL, which was determined to be undecidable.

2 Specifying the disjoint nature of roles

The semantics of DL constructors is defined in terms of an interpretation $I = (\Delta^I, \cdot^I)$ that consists of a non-empty domain Δ^I and an interpretation function $\cdot^I$. The interpretation maps individual names (e.g., *x*, *y* and *z*) into objects or individuals of the domain; and the role names (e.g., *R1* and *R2*) into subsets of the Cartesian product of the domain (Δ^I X Δ^I) as shown in (1) ~ (5).

$$\mathbf{x^I \in \Delta^I} \tag{1}$$
$$\mathbf{y^I \in \Delta^I} \tag{2}$$
$$\mathbf{z^I \in \Delta^I} \tag{3}$$
$$\mathbf{R1^I \subseteq \Delta^I \times \Delta^I} \tag{4}$$
$$\mathbf{R2^I \subseteq \Delta^I \times \Delta^I} \tag{5}$$

Four types of role disjointness can now be distinguished as follows: 1) if an object appears as a range element in role *R1* then it cannot appear as a range element in *R2*; 2) if an object appears as a domain element in role *R1* then it cannot appear as a domain element in *R2*; 3) the conjunction of the conditions in 1 and 2; and 4) two roles can have no instances in common.

For the scenario in which two disjoint roles cannot have instances that have a common range object, the required semantics are shown in (6). A new construct ($|_r$) is proposed to capture the semantics of (6). The disjoint nature of *R1* and *R2* can now be specified as in (7). This constraint would not allow the same object to appear as the range in instances of both roles.

$$\mathbf{\forall x \forall y \forall z\ (x^I, y^I) \in R1^I \Rightarrow (z^I, y^I) \notin R2^I} \tag{6}$$
$$\mathbf{R1\ |_r\ R2} \tag{7}$$

For the scenario in which two disjoint roles cannot have instances that have a common domain object, the required semantics are shown in (8). A new construct ($|_d$) is proposed to capture the semantics of (8). The disjoint nature of *R1* and *R2* can now be specified as in (9). This constraint does not permit the same object to appear as the domain in instances of both the roles *R1* and *R2*.

$$\forall x \forall y \forall z\ (x^I, y^I) \in R1^I \Rightarrow (x^I, z^I) \notin R2^I \quad (8)$$

$$R1\ |_d\ R2 \quad (9)$$

The semantics of two disjoint roles such that no domain element in *R1* can appear as a domain element in *R2* and no range element in *R1* can appear as a range element of *R2* is shown in (10). A new construct (**|**) is defined to capture the semantics of (10). The disjoint nature of *R1* and *R2* can now be specified as in (11).

$$\forall x \forall y \forall z \forall w\ (x^I, y^I) \in R1^I \Rightarrow (x^I, z^I) \notin R2^I \wedge (w^I, y^I) \notin R2^I \quad (10)$$

$$R1\ |\ R2 \quad (11)$$

Finally, each new construct ($|_r$, $|_d$, and **|**) expresses different semantics than rule (12), which states that two roles can have no instances in common as captured in (13). Applying the substitution {z/x} to (6) yields (12) as does applying the substitution {z/y} to (8). Rule (12) is a factor of (10) when applying the substitution {z/y, w/x} to (10). However, it is not possible with the semantics expressed in (12) alone to determine whether the two roles can share domain objects, range objects, or neither as captured by the new constructs.

$$\forall x \forall y\ (x^I, y^I) \in R1^I \Rightarrow (x^I, y^I) \notin R2^I \quad (12)$$

$$R1^I \cap R2^I = \perp \quad (13)$$

3 Conclusions

A premise of this paper is that Semantic Web knowledge representation formalisms should support the declarative representation of property disjointness. Four types of property disjointness have been described in this paper. It should be noted that if a knowledge engineering application required capturing the semantics provided by the constructs $|_r$, $|_d$, and **|**, a workaround could be declaratively achieved, albeit requiring minor re-conceptualization, to enforce the semantics. For example, to achieve $|_d$, the domain of roles *R1* and *R2* would be partitioned into two disjoint concepts and the disjoint nature of roles *R1* and *R2* would then be implied if they were re-defined to use these disjoint concepts as their respective, restricted domains. However, no such workaround appears to exist for expressing $R1 \cap R2 = \perp$ for roles defined on the same domain and range, which suggests that a $DisjointProperties(R_1\ \ldots\ R_n)$ construct may be useful in Description Logics based ontology languages such as OWL. Role intersection constructs are provided by the *ALB* DL [3], which has been proved to be decidable. The analysis of the computational properties of the constructs proposed herein is the subject of ongoing investigation. The investigation will include decidability strategies discussed in [3] and its relevance to the proposed constructs.

References

[1] D. Nardi and R. J. Brachman, “An Introduction to Description Logics,” In F. Baader et al. (Eds.), *The Description Logic Handbook*, Cambridge University Press, 2003.

[2] M. Wessel, “Undecidability of ALC_{RA},” Technical Report No. FBI-HH-M-302/01, Computer Science Department, University of Hamburg, Germany, 2001.

[3] U. Hustadt and R. Schmidt, “Issues of Decidability for Description Logics in the Framework of Resolution,” In R. Caferra and G. Salzer (Eds.), *Automated Deduction in Classical and Non-Classical Logics*, LNAI 1761, Springer, 2000, 192 – 206.