Practical Conforming Datatype Groups

Dave Turner and Jeremy J. Carroll {DavidT,Jeremy.Carroll}@hp.com

HP Laboratories, Bristol, UK

Abstract. The proposed OWL 1.1 language is based on the description logics $\mathcal{SROIQ}$ and $\mathcal{SHOQ}(\mathrm{D_n})$, whose features include the use of n-ary datatype predicates. The means of specifying such predicates, which must be usefully expressive without breaking decidability properties, is omitted; rectifying this omission is nontrivial.

1 Discussion

A finite predicate conjunction over a datatype group $\mathcal{G} = (\Delta_{\mathbf{D}}, \mathbf{D}_{\mathcal{G}}, \Phi^1_{\mathcal{G}}, \Phi_{\mathcal{G}})$ is a statement of the form $\bigwedge_{j=1}^{k} p_j(v_1^{(j)}, \ldots, v_{n_j}^{(j)})$ where p_j is an n_j-ary predicate in $\mathbf{D}_{\mathcal{G}} \cup \Phi^1_{\mathcal{G}} \cup \Phi_{\mathcal{G}}$. The datatype group $\mathcal{G}$ is said to be *conforming* if

1. $\mathbf{D}_{\mathcal{G}}$, $\Phi^1_{\mathcal{G}}$ and $\Phi_{\mathcal{G}}$ are closed under negation,
2. a binary inequality predicate $\neq_d \in \Phi_{\mathcal{G}}$ is defined for each datatype $d \in \mathbf{D}_{\mathcal{G}}$, and importantly
3. the satisfiability of finite predicate conjunctions over $\mathcal{G}$ is decidable.

Pan and Horrocks [1] present a datatype group containing the predicate `kmtrsPerMile` $= D($`k`$,$`m`$,$`"k=1.6*m"`$)$ for converting between distances measured in kilometers and miles. Elsewhere they suggest a predicate of objects that are small enough to have no additional postage cost: `smallObj` $= D($`l`$,$`w`$,$`"l+w<10"`$)$.

Theory of Arithmetic A naïve generalisation of this syntax is too expressive and quickly leads to non-conforming datatype groups. For example, consider a datatype group containing

$$\texttt{integerAddition} = D(\texttt{i}, \texttt{j}, \texttt{k}, \texttt{"i=j+k"})$$
$$\text{and } \texttt{integerMultiplication} = D(\texttt{i}, \texttt{j}, \texttt{k}, \texttt{"i=j*k"})$$

viewed as ternary predicates over $\texttt{integer}^{\mathbf{D}}$. The satisfiability problem over this datatype group amounts to the solution of Diophantine equations, which is Hilbert's tenth problem and is known to be undecidable.

Note that a datatype group containing either one of these predicates could be conforming; it is the presence of both which forces undecidability. The instability of conformingness under merging of intersecting datatype groups is not an artificial problem: in the context of OWL there are a small, finite number of base datatypes and one would expect to want to merge intersecting datatype groups reasonably often. The open-world style of the Semantic Web encourages such merges, so it would be inappropriate to forbid them completely. Furthermore, without both multiplication and addition, end-users would not even be able to convert between °F and °C: `farenheitToCelsius` $= D($`f`$,$`c`$,$`"f=1.8*c+32.0"`$)$.

Implementation of Arithmetic An example from [1] declares that the Yangtze river is 3937.5 miles long and uses the `kmtrsPerMile` predicate to deduce that it is also 6300.0km long, using the XML Schema datatype `float`. If instead we start from a length of 3937.501 miles, then

$$\langle 6300.0015, 3937.501 \rangle \in [\![\texttt{kmtrsPerMile}]\!]$$
$$\text{and } \langle 6300.0015, 3937.5007 \rangle \in [\![\texttt{kmtrsPerMile}]\!],$$

so that the Yangtze river may be deduced also to be 3937.5007 miles long.

We implemented a system to do conversions between `float`s representing lengths in km, m, cm, mm, μm, inches, feet, yards, fathoms, poles, chains, furlongs, statute and nautical miles and leagues, and deduced the length of the Yangtze to be both 6335.3584km and 6361.8555km, and nearly $800,000$ other values, starting from a declaration that its length in miles is 3937.5. These rounding errors were highly dependent on the structure of the definitions of the units, as multiplication in `float` is not associative.

Additionally, suppose the Volga river were declared to be 3668.8003km long, then it would have no value for its `lengthInMile` property at all, since

$$\langle 3668.8000, 2293.0000 \rangle \in [\![\texttt{kmtrsPerMile}]\!]$$
$$\langle 3668.8005, 2293.0002 \rangle \in [\![\texttt{kmtrsPerMile}]\!]$$
$$\text{and } \nexists x \in \texttt{float}.\ 2293.0 < x < 2293.0002$$

Notice that this cannot be remedied by using the arbitrary-precision `decimal` instead of the fixed-precision `float`: for example the temperature 75.0°F has no corresponding `decimal` representation in °C.

In practice, many applications do not require the declarative style of arithmetic that datatypes like `kmtrsPerMile` would allow. Instead, a procedural approach is adequate. For example, a user may be happy that the Volga can be deduced to be 2293.0km long, and may be equally happy with 2293.0002km, as long as one, and only one, of the options is chosen.

2 Conclusions and Future Work

Datatype groups are motivated by the requirements of DL users to be able to express complex constraints simultaneously on multiple data values. However, there has been little discussion regarding datatype groups that satisfy these user requirements whilst also being conforming and computationally feasible. We have shown that this question is far from trivial; see [2] for more details.

References

1. Pan, J.Z., Horrocks, I.: Web Ontology Reasoning with Datatype Groups. In Fensel, D., Sycara, K., Mylopoulos, J., eds.: Proc. of the 2nd International Semantic Web Conference (ISWC2003). (2003)
2. Turner, D., Carroll, J.J.: Practical conforming datatype groups. HP Technical Report HPL-2007-37 (2007)