Higher-Order Dynamic Predicate Logic

Marcos Cramer

University of Bonn
cramer@math.uni-bonn.de

1 Dynamic Predicate Logic

Dynamic Predicate Logic (DPL) [3] is a formalism whose syntax is identical to that of standard first-order predicate logic (PL), but whose semantics is defined in such a way that the dynamic nature of natural language quantification is captured in the formalism:

1. If a farmer owns a donkey, he beats it.
2. PL: $\forall x \forall y\ (farmer(x) \wedge donkey(y) \wedge owns(x,y) \rightarrow beats(x,y))$
3. DPL: $\exists x\ (farmer(x) \wedge \exists y\ (donkey(y) \wedge owns(x,y))) \rightarrow beats(x,y)$

In PL, 3 is not a sentence, since the final occurences of x and y are free. In DPL, a variable may be bound by a quantifier even if it is outside its scope. The semantics is defined in such a way that 3 is equivalent to 2. So in DPL, 3 captures the meaning of 1 while being more faithful to its syntax than 2.

2 Dynamic introduction of function symbols

In natural language mathematical texts, function symbols may be introduced dynamically:

> Suppose that, for each vertex v of K, there is a vertex $g(v)$ of L such that $f(st_K(v)) \subset st_L(g(v))$. Then g is a simplicial map $V(K) \rightarrow V(L)$, and $|g| \backsimeq f$. [4]

Here the natural language quantification "there is a vertex $g(v)$" locally introduces a new vertex to the discourse; but since the choice of the vertex depends on v and we are universally quantifying over v, it globally introduces a function g to the discourse.

If such a dynamic introduction of function symbols is allowed without limitations, this amounts to unrestricted function comprehension, which just like unrestricted set comprehension leads to inconsistencies. So in any formalisation of dynamic function introduction, some limitations have to be enforced. For the sake of simplicity, we use Simple Type Theory (STT) [2] for the formalisation we sketch below. It is also possible to alleviate the limitations set by STT, in order to have a system which has the flexibility and logical strength of ZFC set theory: This can be achieved by adapting Ackermann set theory [1] – a conservative extension of ZFC – to a theory of functions.

3 Typed Higher-Order Dynamic Predicate Logic

We extend DPL to a formalism called *Typed Higher-Order Dynamic Predicate Logic* (THODPL), which allows for dynamically introduced functions, and consequently also for quantification over functions. THODPL has variables typed by the types of STT. In the below examples we use x and y as variables of the basic type i, and f as a variable of the function type $i \to i$. A complex term is built by well-typed application of a function-type variable to an already built term, e.g. $f(x)$ or $f(f(x))$.

The distinctive feature of THODPL syntax is that it allows not only variables but any well-formed terms to come after quantifiers. So 1 is a well-formed formula:

1. $\forall x\ \exists f(x)\ R(x, f(x))$
2. $\forall x\ \exists y\ R(x, y)$
3. $\exists f(\forall x\ R(x, f(x)))$

The semantics of THODPL is to be defined in such a way that 1 has the same truth conditions as 2. But unlike 2, 1 dynamically introduces the function symbol f to the context, and hence turns out to be equivalent to 3.

We now sketch how these desired properties of the semantics can be achieved. Just as in the case of DPL semantics, we define the interpretation $[\![\varphi]\!]$ of a formula φ to be a set of pairs of assignments (rather than a set of assignments as in the case of PL).

In PL and DPL, an assignment assigns objects of the domain D to variables. In THODPL, an assignment assigns elements of D to variables of type i, functions from D to D to variables of type $i \to i$ etc. Additionally, an assignment can also assign an object (or function) to a complex term. For example, the second element of any assignment pair in the interpretation of $\exists f(x)\ R(x, f(x))$ has to assign some object to $f(x)$.

The semantics of most logical connectives is similar to that of DPL, with the exception of $\to$ and $\forall$. We describe – in a somewhat simplified way – the semantics of $\forall$, which plays a crucial role for the equivalence between 1 and 3:

> $[\![\forall x\varphi]\!] = \{\langle g, h\rangle | h$ differs from g in at most some function symbols $f_1, \ldots, f_n$, and for all k that differ from g in at most x, there is an assignment j such that $\langle k, j\rangle \in [\![\varphi]\!]$ and $j(f_i(x)) = h(f_i)(k(x))$ for $1 \leq i \leq n\}$

References

1. Ackermann, W.: Zur Axiomatik der Mengenlehre, Mathematische Annalen, 1956, Vol. 131, pp. 336–345.
2. Church, A.: A formulation of the simple theory of types, J Symbolic Logic, 1940, 5: 5668
3. Groenendijk, J., Stokhof, M.: Dynamic Predicate Logic. Linguistics and Philosophy, Vol. 14, no. 1, 1991, pp. 39-100.
4. Lackenby, M.: Topology and Groups (2008), Lecture Notes, `http://people.maths.ox.ac.uk/lackenby/tg050908.pdf`.