

# TPTP Format for Finite Models

Koen Claessen, Geoff Sutcliffe

A proposal for representing the output from automatic model finding systems - “finite interpretations” - is presented. The goals are:

- Use the TPTP syntax with as little modification as possible
- Use FOF formulae to represent all aspects
- The representation that is amenable to checking and use
- The representation is compact, and multiple interpretations can be easily represented

The following form is proposed:

```
fof(name_of_model, domain, ! [X] : ( X = de1 | X = de2 | ... | X = deN ) ).
```

so that the **deI** terms are an exhaustive list of the domain elements, which are freely chosen constants that do not occur in the signature being interpreted. The annotated formula type **domain** will be added to the TPTP syntax. This representation mixes the language of the formulae being interpreted (over whose terms the **X** ranges) with the language of the interpretation (the **de** terms). To separate these two in your mind, think of each equality literal as meaning “**X** is equal to a term that is interpreted as the domain element **deI**” (we considered writing  $X = \$term(deI)$ , where  $\$term$  would be a new interpreted function in the TPTP language, but that seemed too klunky). To make the domain elements distinct, "double quoted" terms should be used (their distinctness is specified as part of the TPTP syntax). If "double quoted" terms appear in the interpreted signature, then those terms need to be interpreted as the syntactically same domain elements. If the domain elements are not "double quoted" then either equality can be interpreted to specify the inequality of the elements, or multiple interpretations are being represented. The use of multiple **domain** formulae in a file, where the domain elements are not "double quoted", permits domain elements to have alternative names (and the size of the domain is no larger than the maximal number of literals in such a formula).

The following forms are proposed for the function and predicate interpretations:

<pre>fof(name_of_model, meaning,   ( f(de1,...,deM) = deR     &amp; f(de1,...,deP) = deS     ...   ) ).</pre>	<pre>fof(name_of_model, meaning,   ( p(de1,...,deM)     &amp; ~ p(de1,...,deP)     ...   ) ).</pre>
---	---

The annotated formula type **meaning** will be added to the TPTP syntax. If any function or predicate is not interpreted explicitly for a given argument pattern, then all possibilities are intended, thus representing multiple interpretations. It is recommended that no representation be contradictory, which corresponds to no interpretations.

It is recommended that interpretations follow a standard layout, as illustrated by the examples above. However, the conjuncts of function and predicate interpretations can also be separated into individual annotated formulae, and compact forms are possible using more complex formulae, e.g.,

```
fof(name_of_model, meaning, ! [X] : ( X != deK => f(X) = deL ) ).
```

The proposed format leads at least to an easy approach to verifying that the interpretation is a model of a set of formulae - the interpretation is simply adjoined to the formulae, and a trusted model finder is then used to find a model of the combined formula set.