A Model Representation for Infinite Models

Alexander Fuchs

We have currently almost no experience with the TPTP format. Although our theorem prover Darwin,[1] an implementation of the Model Evolution calculus, is participating for the second time in the CASC competition this year, it does not understand the format, but does instead rely on an external conversion tool. For this, but mostly for a more standardized system output, we plan to add TPTP support in the near future.

Our additional interest would then be a convenient way to describe the models generated by our system. These can currently be output either as a *context*, the representation used in the Model Evolution calculus, or equivalently as a *DIG* (disjunctions of implicit generalizations) representation,[2] in formats custom to our prover.

These are finite representations of possibly infinite first-order models, by specifying hierarchies of default interpretations and exceptions. A context is a set of literals, where by default each positive literal L makes all its instances true in the interpretation, except for those which are also instances of a negative literal more specific than L (simplified description). In contrast a DIG consists of a set of implicite generalizations, positive atoms with a set of negative atoms as exceptions. For example, the model specified by $\{p(x, y) \mid (\neg(x = a) \wedge \neg(y = a)) \vee ((x = a) \wedge (y = a))\}$, could be represented as the context $\{p(x, y), \neg p(x, a), \neg p(a, x), p(a, a)\}$ or the DIG $\{p(x, y) \setminus \{p(x, a), p(a, x)\}, p(a, a)\}$.

I am not too sure about the details of the statement of Claessen and Sutcliffe, but it seems like the above DIG could be encoded in their proposal more or less as:

```
% map each symbol to 'itself':
fof(name_of_model,domain, ! [X]    : ( X = "f" | X = "a" | ...) ).

% specify each implicit generalization:
fof(name_of_model,meaning,! [X, Y] :
    ( ~(X = "a") & ~(Y = "a") ) => p(X, Y) ).
fof(name_of_model,meaning,! [X, Y] :
    $true                        => p(a, a) ).
```

[1] http://goedel.cs.uiowa.edu/Darwin

[2] for an introduction and comparison see Fermueller, Pichler, "Model Representation via Contexts and Implicit Generalizations"