MSPASS: Subsumption Testing with SPASS

Ullrich Hustadt\(^1\) and Renate A. Schmidt\(^1\) and Christoph Weidenbach\(^2\)

\(^1\)Department of Computing, Manchester Metropolitan University,
Chester St., Manchester M1 5GD, United Kingdom
{U.Hustadt, R.A.Schmidt}@doc.mmu.ac.uk

\(^2\)Max-Planck-Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany
weidenb@mpi-sb.mpg.de

MSPASS is an enhancement of the first-order theorem prover SPASS \([17; 18; 19; 20]\) with a translator of modal formulae, formulae of description logics, and formulae of the relational calculus into first-order logic with equality. SPASS is one of the fastest and most sophisticated existing first-order theorem provers and its performance compares well with special purpose theorem provers for modal logics, description logics and first-order logic, in general \([7; 9; 16]\).

Input language
The input language of SPASS was extended to accept, in addition to first-order formulae, also modal, relational and description logic formulae as input. Description logic and modal formulae are built from a vocabulary of propositional symbols of two disjoint types, namely, type Boolean (concept) and type relational (role). These need to be declared in advance as predicates with zero arity. The repertoire of logical constructs includes:

- the standard relational translation method (which is determined by the usual semantics of the language),
- the functional translation method \([1; 4; 13]\) and
- the optimised functional translation method \([14]\) including a variation defined in terms of \(n\)-ary predicates instead of unary predicates and paths \([6; 7]\), as well as
- the semi-functional translation method \([10; 11]\).

In the current implementation the standard relational translation method is most general and applies to the language described above. Some restrictions apply to the other methods. The functional translation method and semi-functional translation methods are available only for ALC, possibly with serial (total) roles, plus ABox statements, terminological axioms and general inclusion and equivalence axioms. The optimised functional translation methods are implemented only for ALC, possibly with serial roles.

Features of SPASS
SPASS is a sound and complete theorem prover for first-order logic with equality. It is an implementation of a saturation-based resolution and superposition calculus with simplification \([2; 3]\). In particular,

- it uses ordered resolution, and ordered superposition with selection,
- it supports splitting and branch condensing (splitting amounts to case analysis while branch condensing resembles branch pruning or backjumping),
- it has an extensive set of reduction and simplification rules,
- and it supports dynamic sort theories by additional inference and reduction rules.

Of particular importance for satisfiable formulae are ordered inference, splitting, and condensing while for randomly generated formulae unit propagation, and branch condensing are important as well. Part of SPASS is a fast converter of first-order formulae into clausal form with special features such as optimised and strong Skolemisation, and an improved implementation of renaming \([12]\).

Translation mappings
A number of different translation methods are available in MSPASS. These include:

- the standard relational translation method (which is determined by the usual semantics of the language),
- the functional translation method \([1; 4; 13]\) and
- the optimised functional translation method \([14]\) including a variation defined in terms of \(n\)-ary predicates instead of unary predicates and paths \([6; 7]\), as well as
- the semi-functional translation method \([10; 11]\).

The translation mapping into clausal form with special features such as optimised and strong Skolemisation, and an improved implementation of renaming \([12]\).
On termination, SPASS does not only produce a 'yes'/no' answer, but it also outputs a proof or a saturated set of clauses depending on whether the input problem is unsatisfiable or not. A finite saturated set of clauses provides a characterisation of a class of models for the input problem.

For certain logics and using certain flag settings MSPASS is known to provide decision procedures. More specifically:

- For the relational translation of description logics below ALC and including ALC [8]. ALC is the extension of ACC with role formulae expressed in terms of the top and bottom role, conjunction, disjunction, negation and converse.
- For the optimised functional translation of ACC with or without serial, reflexive or transitive roles [15].
- For the semi-functional translation of ACC with or without serial, reflexive, symmetric or transitive roles [5].

Implementation

Both SPASS and MSPASS are implemented in ANSI C. The translation routines of MSPASS have alpha status (there are plans to change the input syntax slightly).

Performance

Extensive testing has been done with different versions of SPASS but as yet not with MSPASS. The tests which have been performed and published thusfar have been for ACC or multi-modal $K_{[n]}$ with or without reflexive and transitive modalities [6; 7; 9]. In all cases the translation employed was the optimised functional translation in terms of $n$-ary predicates.

Availability

SPASS as well as MSPASS are available from http://spass.mpi-sb.mpg.de/ where also interactive use of the systems is possible.

References