SPASS & FLOTTER
Version 0.42

Christoph Weidenbach and Bernd Gaede and Georg Rock

Max-Planck-Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany,
email: weidenb@mpi-sb.mpg.de, phone: +49-681-9325221

System Description

FLOTTER¹ and SPASS² are a clause normal form translator for first-order
logic and a theorem prover for first-order logic with equality, respectively. The
algorithm implemented by FLOTTER includes an improved form of the re-
naming technique developed by Thierry Boy de la Tour [2] and the optimized
Skolemization technique by Ohlbach and Weidenbach [7].

SPASS is based on the superposition calculus developed by Bachmair and
Ganzinger [1] extended with the sort techniques developed by Weidenbach [14,
15]. The superposition calculus relies on a reduction ordering which is total on
ground terms. For SPASS we chose an extension of the Knuth-Bendix ordering
that was first presented by Peterson [10]. In addition to the usual inference,
deletion and simplification rules, SPASS includes a rule for case analysis, called
splitting.

1 SPASS

1.1 Calculus

A clause in SPASS has the form

\[
sort \text{ constraint atoms } \ | \ | antecedent \text{ atoms } \rightarrow \ text{ succedent atoms}
\]

with the semantics that the conjunct of the sort constraint and antecedent atoms
logically implies the disjunct of the succedent atoms.

The inference rules equality resolution, equality factoring, superposition left
and superposition right are applied to the antecedent and succedent atoms with
respect to the usual ordering and selection restrictions [1]. However, their appli-
cation is limited to clauses with a “solved” sort constraint. The sort constraint
represents the sort restrictions on the variables. There are two hyper resolution
like inference rules for transforming sort constraints into solved form: Sort res-
olution and empty sort. These rules implement a specific strategy of the sorted

¹ FLOTTER means “faster” in German.
² Synergetic Prover Augmenting Superposition with Sorts, SPASS means “fun” in
German.
unification algorithm [15] on the sort constraint atoms. In addition to these inference rules, SPASS includes a splitting rule. The splitting rule is a variant of the usual \( \beta \)-rule of tableau. If SPASS splits a clause into two different cases, the two parts do not share variables, i.e., these parts can be independently refuted.

The implementation of SPASS includes powerful deletion and simplification rules: tautology deletion, subsumption, condensing, an efficient variant of contextual rewriting, sort simplification and clause deletion. The latter two rules are based on the sort information contained in the clauses. Sort simplification implements the deletion of constraint literals that are well-sorted. Whenever an atom of the form \( S(t) \) occurs in the sort constraint such that \( t \) is of sort \( S \) with respect to the sort information contained in the problem, the atom \( S(t) \) is deleted. This problem is always decidable in polynomial time. Clause deletion deletes clauses where the sort constraint can be shown to be unsolvable. In general, this rule demands the approximation of the sort information contained in the clauses to get a decidable solvability problem for sort constraints.

A more detailed explanation of the calculus is contained in the SPASS distribution (see Section 4).

1.2 Control

The proof search is controlled by two sets of clauses: The set of worked-off clauses and the set of usable clauses. At the beginning all input clauses are in the usable set. Then a clause is selected from the usable set and moved to the worked-off set. Clauses are selected in a fair way according to their weight (the number of symbols) and depth in the search space. All inference rules consider a selected clause and the set of worked-off clauses to infer new clauses. The new clauses are processed by the deletion and simplification rules and, if they are not redundant, are finally inserted into the usable set. The algorithm continues by selecting the next clause from the usable set.

Besides this loop, SPASS performs a depth first, left to right tree search according to the results of the splitting rule. If an empty clause is derived, it may not be valid at top level and only refute some branch of the tree. Then this branch is closed and SPASS continues its search at the next open branch of the tree.

2 FLOTTER

The CNF algorithm implemented by FLOTTER basically consists of the steps described below. For a more detailed description, see the SPASS distribution (Section 4) or Georg Rock's master thesis [11].

1. Rename subformulæ of the input formula in order to obtain short clause normal forms. This step is explained in more detail in the next paragraph.
2. Remove implications using the usual transformation laws. Equivalences are removed with respect to their polarity.