A MECHANICAL SOLUTION OF SCHUBERT'S STEAMROLLER BY MANY-SORTED RESOLUTION

Christoph Walther
Institut für Informatik I
UNIVERSITÄT KARLSRUHE
Postfach 6380
D-7500 Karlsruhe 1

Abstract We demonstrate the advantage of using a many-sorted resolution calculus by a mechanical solution of a challenge problem. This problem known as "Schubert's Steamroller" had been unsolved by automated theorem provers until now. Our solution clearly demonstrates the power of a many-sorted resolution calculus. The proposed method is applicable to all resolution-based inference systems.

1. SCHUBERT'S PROBLEM

In 1978, *L. Schubert* of the University of Alberta set up the following challenge problem:

> Wolves, foxes, birds, caterpillars, and snails are animals, and there are some of each of them. Also there are some grains, and grains are plants. Every animal either likes to eat all plants or all animals much smaller than itself that like to eat some plants. Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves. Wolves do not like to eat foxes or grains, while birds like to eat caterpillars but not snails. Caterpillars and snails like to eat some plants. Therefore there is an animal that likes to eat a grain-eating animal.

This problem became well known since in spite of its apparent simplicity it turned out to be too hard for existing theorem provers because the search space is just too big.

Using the following predicates as an abbreviation:

A(x) - x is an animal	W(x) - x is a wolf
F(x) - x is a fox	B(x) - x is a bird
C(x) - x is a caterpillar	S(x) - x is a snail
G(x) - x is a grain	P(x) - x is a plant
M(xy) - x is much smaller than y	
E(xy) - x likes to eat y	

we obtain the following set of clauses as a predicate logic formulation of the problem:

(1) $\{W(w)\}$ (2) $\{F(f)\}$
(2) $\{B(b)\}$ (4) $\{C(c)\}$
(5) $\{S(s)\}$ (6) $\{G(g)\}$
(7) $\{\bar{W}(x_1), A(x_1)\}$ (8) $\{\bar{F}(x_1), A(x_1)\}$
(9) $\{\bar{B}(x_1), A(x_1)\}$ (10) $\{\bar{C}(x_1), A(x_1)\}$
(11) $\{\bar{S}(x_1), A(x_1)\}$ (12) $\{\bar{G}(x_1), P(x_1)\}$
(13) $\{\bar{A}(x_1), \bar{P}(x_2), \bar{A}(x_3), \bar{P}(x_4), E(x_1x_2), \bar{M}(x_3x_1), \bar{E}(x_3x_4), E(x_1x_3)\}$
(14) $\{\bar{C}(x_1), \bar{B}(x_2), M(x_1x_2)\}$ (15) $\{\bar{S}(x_1), \bar{B}(x_2), M(x_1x_2)\}$
(16) $\{\bar{B}(x_1), \bar{F}(x_2), M(x_1x_2)\}$ (17) $\{\bar{F}(x_1), \bar{W}(x_2), M(x_1x_2)\}$
(18) $\{\bar{F}(x_1), \bar{W}(x_2), \bar{E}(x_2x_1)\}$ (19) $\{\bar{G}(x_1), \bar{W}(x_2), \bar{E}(x_2x_1)\}$
(20) $\{\bar{B}(x_1), \bar{C}(x_2), E(x_1x_2)\}$ (21) $\{\bar{B}(x_1), \bar{S}(x_2), \bar{E}(x_1x_2)\}$
(22) $\{\bar{C}(x_1), P(h(x_1))\}$ (23) $\{\bar{C}(x_1), E(x_1h(x_1))\}$
(24) $\{\bar{S}(x_1), P(i(x_1))\}$ (25) $\{\bar{S}(x_1), E(x_1i(x_1))\}$
(26) $\{\bar{A}(x_1), \bar{A}(x_2), G(j(x_1x_2))\}$
(27) $\{\bar{A}(x_1), \bar{A}(x_2), \bar{E}(x_1x_2), \bar{E}(x_2j(x_1x_2))\}$

where w,f,b,c,s and g are skolemconstants, x_1, x_2, x_3 and x_4 are universally quantified variables and h, i and j are skolemfunctions.

Figure 1.1 Schubert's problem in clause notation

In the fall of 1978 L. Schubert spend his sabbatical at the University of Karlsruhe and a first-order axiomatization of his Problem was given to the *Markgraf Karl Refutation Procedure* (MKRP) [BES81], a resolution-based automated theorem prover under development at the University of Karlsruhe. The system generated the clause set of figure 1.1, but failed to find a refutation. Though several significant improvements have been incorporated into the MKRP-system during the last six years, it is still unable to find a refutation of the above clause set today. The same is true for all the other automated theorem provers we know about, which were confronted with this problem. But there exists a refutation as it can be seen from Schubert's hand computed deduction of the empty clause [Sch78,Wal84a]. Looking at the clause set of figure 1.1 and the handcomputed refutation of the problem, the reason for the difficulties of an automated theorem prover in computing a solution become apparent:

* The size of the initial search space (we can compute 102 distinct clauses, 94 resolvents and 8 factors already in the first generation) and
* the search depth necessary to compute the empty clause (which is 20 in Schubert's handcomputed solution)

leads to such a

* rapidly growing search space that the time and/or space boundaries of an automated theorem prover are exceeded before the empty clause can be deduced.

This holds true even if we use some refinements, like for instance *set-of-support* [WRC65], which reduces the initial search space to 28 potential resolvents and 2 potential factors.

2. A MANY-SORTED SOLUTION

The first-order axiomatization in figure 1.1 reflects a specific view of the given problem: We consider an unstructured universe, the objects of which are associated with properties (expressed by unary predicates) - for instance "is a wolf", "is an animal", "is a grain" etc. - and where relations between these properties are given by implications.

But there is another, more natural way of looking at the given scenario, which, incidentally, enables a human to find a solution: Given a *many-sorted* universe, which consists of sorts of objects like wolves, animals, grains, plants etc. and certain relations between these objects, e.g. wolves *are* animals and grains *are* plants, everything which is true for animals (or plants), automatically holds for wolves (or grains respectively). In this scenario we talk about the preferences of *wolves* of eating *grains* and not about these preferences of *all* objects, which satisfy "is a wolf" and "is a grain".

Hence a many-sorted first-order calculus is more suitable for a formalization of Schubert's problem. In such a calculus the domains and ranges of functions, predicates and variables are restricted to certain subsets of the universe (which are given as a *hierarchy of sorts*) where these restrictions are respected by the inference rules. In a many-sorted axiomatization the problem reads (in clause notation) as follows:

(1) *type* w:W	(2) *type* f:F
(3) *type* b:B	(4) *type* c:C
(5) *type* s:S	(6) *type* g:G
(7) *sort* W<A	(8) *sort* F<A
(9) *sort* B<A	(10) *sort* C<A
(11) *sort* S<A	(12) *sort* G<P
(13) $\{E(a_1p_1),\bar{M}(a_2a_1),\bar{E}(a_2p_2),E(a_1a_2)\}$	
(14) $\{M(c_1b_1)\}$	(15) $\{M(s_1b_1)\}$
(16) $\{M(b_1f_1)\}$	(17) $\{M(f_1w_1)\}$
(18) $\{\bar{E}(w_1f_1)\}$	(19) $\{\bar{E}(w_1g_1)\}$
(20) $\{E(b_1c_1)\}$	(21) $\{\bar{E}(b_1s_1)\}$
(22) *type* h(C):P	(23) $\{E(c_1h(c_1))\}$
(24) *type* i(S):P	(25) $\{E(s_1i(s_1))\}$
(26) *type* j(AA):G	(27) $\{\bar{E}(a_1a_2),\bar{E}(a_2j(a_1a_2)\}$

Figure 2.1 The many-sorted version of Schubert's problem in clause notation

In this axiomatization the symbols W,F,B,C, S,A,G and P are used as *sort symbols* which are ordered by the *subsort order* according to the subsort declarations (7) - (12), i.e.

W,F,...,S are subsorts of A and G is a subsort of P. The type declarations (1) - (6), e.g. *type* w:W, define a *signature* in which for instance w is a constant of sort W.

The type declarations (22), (24) and (26) denote an extension of the given signature computed by the system for the skolem-functions h, i and j, e.g. h is an unary function of sort P with domainsort C. The subscripted lower case letters, e.g. a_1, a_2, $p_1 \ldots$, are universally quantified variables of the sort denoted by the corresponding upper case letter, e.g. A,P,... .

The MKRP-system was extended to a many-sorted theorem prover on the basis of the many-sorted calculus as proposed in [Wal83]. In this calculus, the subsort order and the signature cause a *restriction* of the unification procedure [Wal84b]: A variable x can only be unified with a term t iff the sort of t (which is determined as the sort of the outermost symbol of t) is a subsort of or equals the sort of x. For instance we can resolve upon the literals 20(1) and 27(1) in figure 2.1 using the most general unifier $\{a_1 \leftarrow b_1, a_2 \leftarrow c_1\}$ (but not $\{b_1 \leftarrow a_1, c_1 \leftarrow a_2\}$).

However there is no such resolvent upon the literals 20(1) and 21(1) in the many-sorted resolution calculus since there is no subsort relation between C and S. As a consequence the variables c_1 and s_1 are not unifiable.

Using the clause set of figure 2.1 the MKRP-system computed the following refutation within 10 resolution steps:

(28)	$\{E(a_1p_1), \bar{M}(a_2a_1), \bar{E}(a_2j(a_1a_2))\}$	;13(4) + 27(1)
(29)	$\{E(w_1p_1), \bar{E}(f_1j(w_1f_1))\}$	;17(1) + 28(2)
(30)	$\{\bar{E}(f_1j(w_1f_1))\}$	;19(1) + 29(1)
(31)	$\{E(f_1p_1), \bar{E}(b_1j(f_1b_1))\}$	;16(1) + 28(2)
(32)	$\{\bar{E}(b_1j(f_1b_1))\}$	;30(1) + 31(1)
(33)	$\{E(b_2p_1), \bar{M}(s_1b_2), \bar{E}(s_1p_2)\}$	;13(4) + 21(1)
(34)	$\{\bar{M}(s_1b_1), \bar{E}(s_1p_1)\}$	;32(1) + 33(1)
(35)	$\{\bar{E}(s_1p_1)\}$	;15(1) + 34(1)
(36)	$\{\ \}$	;25(1) + 35(1)

continue next page

Figure 2.2 The MKRP-solution of the many-version of Schubert's problem

For this proof the system uses the replacement principle [Rob65] (cf. clause 28) and the set-of-support strategy [WRC65] with clause 27 as the set of support. Having computed the 5th resolvent, i.e. clause 32, the control of the search was taken over by the terminator module [AO83], which had found a unit-refutation for the remaining clause set.

But why does the system find a solution for the many-sorted formulation, when it did not find one for the unsorted type? The reason is the significantly reduced search space as comparted to the clause set of figure 1.1: For the many-sorted case there are only 12 clauses with 16 literals instead of 27 clauses with 65 literals.

The resulting search space is further reduced by the constraints imposed on the unification procedure: For instance we can compute the resolvent upon the literals 20(3) and 21(3) in figure 1.1 yielding $\{\bar{B}(x_1), \bar{C}(x_2), \bar{S}(x_2)\}$ from which we obtain $\{\bar{C}(x_2), \bar{S}(x_2)\}$ by resolution with clause 3. But $\bar{C}(x_2)$ [$\bar{S}(x_2)$] can only be resolved upon 4(1) [5(1)] yielding a pure clause $\bar{S}(c)$ [$\bar{C}(s)$] in either step. In the many-sorted case these *deadends are impossible*: the corresponding resolution step upon the literals 20(1) and 21(1) in figure 2.1 is blocked because the variables c_1 and s_1 are not unifiable.

As a result the size of the initial search space is totally reduced to 12 potential resolvents (compared to 94 potential resolvents and 8 potential factors), which again can be reduced to 3 potential resolvents (compared to 28 plus 2 potential factors) if the set-of-support strategy is used. The following diagram compares the statistical values of both solutions, where the values of the handcomputed solution are given in the black boxes. The relation between the size of the corresponding boxes is proportional to the ratio of the values:

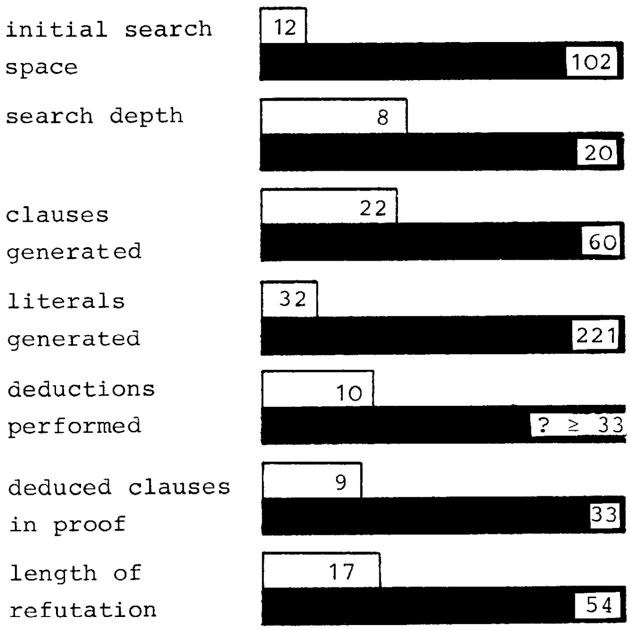

Figure 2.3 The statistical values of both solutions

3. THE GENERAL SOLUTION

Having found a solution of a many-sorted version of Schubert's steamroller, we have to verify that this solution also solves the original problem.

It is well known how to compare a many-sorted calculus with its unsorted counterpart by so-called *sort axioms* and *relativizations* (cf. [Obe62, Wal83]): The sort axioms serve to express the signature and the subsort order in terms of first-order formulas (viz. implications). The relativization of a formula expresses the sort of each variable by atomic formulas using sort symbols as unary predicates.

In clause notation we obtain for instance clause 1 of figure 1.1 as the sort axiom corresponding to the type declaration 1 of figure 2.1 and we obtain clause 7 of figure 1.1 as the sort axiom for the subsort declaration 7 of figure 2.1. The relativization of a clause is obtained by extending the clause with all literals of form $\bar{Q}(x)$, where x is a variable of sort Q in the given clause . For instance clause 13 of figure 1.1 is a relativization of clause 13 in figure 2.1.

Defining S as the set of all clauses of figure 2.1, $\hat{S}$ as the set of all relativized clauses of S and A^{Σ} as the set of all sort axioms for the signature and the subsort order defined in figure 2.1, it is easily verified that $(\hat{S} \cup A^{\Sigma})$ is the set of all clauses of figure 1.1 (up to variable renamings). From the Soundness-, the Completeness- and the Sort-Theorem for the many-sorted resolution calculus [Wal83] we obtain

$$S \vdash_{\Sigma} \Box \text{ iff } (\hat{S} \cup A^{\Sigma}) \vdash \Box$$

(where $\vdash_{\Sigma} \Box$ denotes a refutation in the many-sorted calculus and $\vdash \Box$ denotes a refutation in the ordinary resolution calculus). Moreover one direction of this equivalence is *constructive*, i.e. there exists an algorithm which translates each refutation of S into a refutation of $(\hat{S} \cup A^{\Sigma})$. Hence by solving the many-sorted version of Schubert's problem, a solution of the original problem is also obtained using the above transformations.

4. CONCLUSION

Most mathematical problems have a many-sorted structure and it is not a mere accident that almost all mathematical textbooks are written in a many-sorted language (albeit often very implicit).

The advantage of many-sorted theorem proving was also recognized by [Hay71, Hen72, Wey77, Cha78, BM79, Coh83]. Many-sorted first-order calculi were investigated by [Her30, Sch38, Sch51, Wan52, Hai57, Gil58, Obe62, Ide64] and [Wal83] extends the results to the resolution calculus with paramodulation.

The advantage of this calculus for automated theorem proving was demonstrated here using Schubert's steamroller. Of course, the real power of a many-sorted theorem prover is only obtained, if the problem to be solved has a *many-sorted structure*: It turned out in several example runs (cf. [Wal83]) that the performance of the system increases with an increasing cardinality of the subsort order relation.

Often problems with a many-sorted structure are presented in an unsorted axiomatization. For such problems an algorithm has been developed which translates an unsorted axiomatization into an equivalent many-sorted axiomatization [Sch84].

Acknowledgement

I would like to thank *J. Siekmann* for his helpful criticism and support which greatly contributed to the present form of this paper.

References

[AO83] *Antoniou, G. and H.J. Ohlbach* Terminator. Proc. of the 8th Intern. Joint Conference on Artificial Intelligence (IJCAI-83), Karlsruhe (1983)

[BES81] *Bläsius, K., Eisinger, N., Siekmann, J., Smolka, G., Herold, A., and C. Walther* The Markgraf Karl Refutation Procedure. Proc. of the 7th International Joint Conf. on Artificial Intelligence (IJCAI-81), Vancouver (1981)

[BM79] *Boyer, R.S. and J S. Moore* A Computational Logic. Academic Press (1979)

[Cha78] *Champeaux, D. de* A Theorem Prover Dating a Semantic Network. Proc. of AISB/GI Conf., Hamburg (1978)

[Coh83] *Cohn, A.G.* Improving the Expressiveness of Many-Sorted Logic. Proc. of the 3rd Nat. Conf. on Artificial Intelligence (AAAI-83), Washington (1983)

[Gil58] *Gilmore, P.C.* An Addition to "Logic of Many-Sorted Theories". Compositio Mathematica 13 (1958)

[Hai57] *Hailperin, T.* A Theory of Restricted Quantification I. The Journal of Symbolic Logic 22 (1957)

[Hay71] *Hayes, P.* A Logic of Actions. Machine Intelligence 6, Metamathematics Unit, Univ. of Edinburgh (1971)

[Hen72] *Henschen, L.J.* N-Sorted Logic for Automatic Theorem Proving in Higher-Order Logic. Proc. ACM Conference, Boston (1972)

[Her30] *Herbrand, J.* Recherches sur la théorie de la demonstration (Thèse Paris), Warsaw (1930) chapter 3. Also in "Logical Writings" (W.D. Goldfarb ed.), D.Reidel Publ.Co.(1971)

[Ide64] *Idelson, A.V.* Calculi of Constructive Logic with Subordinate Variables. American Mathematical Society Translations (2) 99 (1972) - translation of Trudy Mat. Inst. Steklov 72 (1964)

[Obe62] *Oberschelp, A.* Untersuchungen zur mehrsortigen Quantorenlogik. Mathematische Annalen 145 (1962)

[Rob65] *Robinson, J.A.* A Machine-Oriented Logic Based on the Resolution Principle. JACM 12 (1965), also in [SW83]

[Sch38] *Schmidt, A.* Über deduktive Theorien mit mehreren Sorten von Grunddingen. Mathematische Annalen 115 (1938)

[Sch51] *Schmidt, A.* Die Zulässigkeit der Behandlung mehrsortiger Theorien mittels der üblichen einsortigen Prädikatenlogik. Mathematische Annalen 123 (1951)

[Sch78] *Schubert, L.* Private Communication

[Sch84] *Schmidt-Schauss, M.* Mechanical Generation of Sorts in Clause Sets. Interner Bericht, Fachber. Informatik, Universität Kaiserslautern (forthcoming 1984)

[SW83] *Siekmann, J. and G. Wrightson (eds.)* Automation of Reasoning - Classical Papers on Computational Logic, vol 1, Springer-Verlag (1983)

[Wal83] *Walther, C.* A Many-Sorted Calculus based on Resolution and Paramodulation. Proc. of the 8th International Joint Conference on Artificial Intelligence (IJCAI-83), Karlsruhe (1983)

[Wal84a] *Walther, C.* Schubert's Steamroller - A Case Study in Many-Sorted Resolution. Interner Bericht 5/84, Institut für Informatik I, Universität Karlsruhe (1984)

[Wal84b] *Walther, C.* Unification in Many-Sorted Theories. Proc. of the 6th European Conf. on Artificial Intelligence (ECAI-84), Pisa (forthcoming 1984)

[Wan52] *Wang, H.* Logic of Many-Sorted Theories. The Journal of Symbolic Logic 17 (1952)

[Wey77] *Weyhrauch, R.W.* FOL A Proof Checker for First-Order Logic. MEMO AIM-235.1, Stanford Artificial Intelligence Laboratory, Stanford University (1977)

[WRC65] *Wos, L.T., Robinson, G.A. and D.F. Carson* Efficiency and Completeness of the Set of Support Strategy in Theorem Proving. JACM 12 (1965), also in [SW83]