A MANY-SORTED CALCULUS BASED ON RESOLUTION AND PARAMODULATION

Christoph Walther

Institut fur Informatik I, Universitat Karlsruhe

Abstract The first-order calculus whose well formed formulas are clauses and whose sole inference rules are factorization, resolution and paramodulation is extended to a many-sorted calculus. As a basis for Automated Theorem Proving, this many-sorted calculus leads to a remarkable reduction of the search space and also to simpler proofs. The soundness and completeness of the new calculus and the Sort-Theorem, which relates the many-sorted calculus to its one-sorted counterpart, are shown. In addition results about term rewriting and unification in a many-sorted calculus are obtained. Practical examples and a proof protocol of an automated theorem prover based on the many-sorted calculus are presented.

"As a rule," said Holmes, "the more bizarre a thing is the less mysterious it proves to be. It is your commonplace, featureless cases which are really puzzling."

A.C. Doyle, The Red-Headed League

7. *Introduction*

Sorts are frequently used in practical applications of the first-order predicate calculus. For example we write formulas like

(i) $\forall x{:}S.\ \Phi(x)$ and $\exists x{:}S.\ \Phi(x)$

and treat them *formally* as *abbreviation** for

(ii) $\forall x.\ S(x) \supset \Phi(x)$ and $\exists x.\ S(x) \wedge \Phi(x)$).

We use *we.ll sorted* formulas because they provide a convenient shorthand notation for ordinary first-order formulas. But sorts also inflence*, the, deduction** from a given set of well sorted formulas. For instance, if P is a predicate only defined on the sort % of integers, we will never perform a deduction like $\forall x{:}\mathbb{Z}.\ P(x) \vdash P(\sqrt{2})$. Proofs are simplified, because a *many-sorted calculus* is more adapted to a *many*- sorted *theory* and hence not surprisingly *deduction* which respects sorts* as well as the *usage of well borted formulas* reflect the everyday usage of predicate logic.

Let us sketch how a *many-sorted* (mehrsortig) calculus is developed from a given (sound and complete) first-order one-sorted (einsortig) calculus: Assume we have a set of sort *symbols,* ordered by a given subsort *order.* Variable and function symbols (of our given calculus) are associated with a certain sort symbol. The *sort of a term* is determined by the sort of its outermost symbol. In the construction of well *Sorted* (sortenrecht) formulas, we only allow well sorted terms of a certain *domainsort* or of a subsort of this domainsort for each argument position of a function or predicate symbol.

The *inference rule** of the many-sorted calculus are the inference rules of the given calculus, but with the restriction that *only well *orted formula** can be deduced by an application of the restricted inference rules. Starting with well sorted formulas this guarantees that only well sorted formulas are derived in a deduction in the many-sorted calculus.

Taking J. *Herbrand** thesis [Her30] as the starting-point, various many-sorted versions of different fir'st-order calculi have been proposed and investigated by A. *Schmidt* [Sch38,Sch51], *H. Wang* [Wan52], *P.C.Gilmore* [Gil58], T. *Hailperin* [Hai57], A. *Oberschelp* [Obe62] and A. V. *Idehon* [Ide64].

The advantages of a many-sorted calculus are well recognized within the field of *Automated Theorem Proving,* e.g. [Hay71 , Hen72]. Several theorem proving programs are also based on some kind of a many-sorted calculus, e.g. [Wey77 ,Cha78 ,BM79](unfortunately without a sound theoretical foundation). As a result of this the works cited above become of practical significance.

Most theorem proving programs are based on a first-order calculus whose inference rules are factorization, resolution and panamadulate on [Rob65, WR73] and whose formulas (called clauses.) are in skolemized conjunctive normal form [Lov78]. We call such a calculus an *RP-calculus.*

In this paper, we shall define the RP-calculus, i.e. a many-sorted version of the RP-calculus, and we shall introduce a notion of E-unsatisfiability of sets of well sorted clauses, called *ΣE-unsatisfiability.*

Obviously we are only interested in a many-sorted calculus which is sound and complete, but in addition we can ask: Which formulas do we expect as theorems of the **ΣRP-calculus** compared to those of the RP-calculus? To facilitate a comparison between both these calculi, we represent the relations between the function symbols and the sort symbols as well as the subsort order by the set A^{Σ} of sort axioms (Sortenaxiome), i.e. a set of first-order formulas. For a well sorted formula Φ, as e.g. (i) above, the relativization $\hat{\Phi}$ (Sortenbeschrankung , Relat vierung) of Φ is the unabbreviated version of Φ, as e.g. (ii), where sort symbols are used as unary predicate symbols to express the sort of a variable symbol. Now the relation between the RP- and the **ΣRP**-calculus can be expressed by the following diagram:

$$\begin{array}{lcl} S \text{ is } \Sigma E\text{-unsatisfiable} & \overset{(1)}{\longleftrightarrow} & S^{E} \vdash_{\Sigma} \Box \\ \quad (2.1)\ \updownarrow & & (2.2)\ \updownarrow \\ \hat{S} \cup A^{\Sigma} \text{ is } E\text{-unsatisfiable} & \underset{(3)}{\longleftrightarrow} & \hat{S}^{E} \cup A^{\Sigma} \vdash \Box \end{array}$$

Here S denotes the extension of a set S of well sorted clauses by all functionally-reflexive axioms **[WR73]** and D denotes the empty clause. $\vdash_{\Sigma}$ denotes a deduction in the ERP-calculus and $\vdash$ stands for a deduction in the RP-calculus. We assume that the set of sort axioms A^{Σ} is in clausal form and we
A
let S denote the set of all relativizations of the well sorted clauses in S.
In the diagram, (1) and (3) express the soundness and completeness of the **ΣRP- and** the RP-calculus respectively. The equivalences (2.1) and (2.2) are called the Sort-Theoicm (Sortensatz), (2.1) is its modclthe.oKe.tic part and (2.2) its proof theroretic. pant.
The Sort-Theorem establishes the connection between both calculi and also shows the advantages we have in using a many-sorted calculus: We obtain a shorter refutation of a smaller set of shorter clauses, when proving $S^{E} \vdash_{\Sigma} \Box$ instead $\hat{S}^{E} \cup A^{\Sigma} \vdash \Box$. e reason is that deductions about sort-relationships, which are performed explicitly in the one-sorted calculus, are built into the inference, mechanism in the many-sorted calculus.

In this paper, we shall show the soundness and completeness of the ERP-calculus (1), as well as the modeltheoretic part of the Sort-Theorem (2.1), i.e. we show that the above diagram is commutative. We shall also consider term rewriting under sorts because important aspects of paramodulation are related to term rewriting.

We shall demonstrate that the ERP-calculus is only complete if it contains a special inference rule, the so-called weakening rule. This rule is specific to a many-sorted calculus, because it cannot be applied if only one sort is given, and hence in our formulation the RP-calculus is but a special case of the **ΣRP-calculus.** We also present special results about unification under sorts which are necessitated by the weakening rule.

The practical application of the **ΣRP**-calculus in Automated Theorem Proving leads to a drastic reduction of the search space as compared to the RP-calculus. This has been demonstrated by an implementation of the ERP-calculus in an existing proof procedure

[BES81,Ohl82].

This paper is a shortened version of the technical report [Wal82a] to which the reader is referred for the omited proofs.

2. Formal Preliminaries

It is assumed that the reader is familiar with the basic terminology of resolution and paramodulation based theorem proving (e.g. [Lov78]).

The RP-Calculus Let T denote the set of all well formed terms over the alphabet of variable *symbols* $\mathfrak{V}$, the alphabet of constant symbols C and the alphabet of function symbols F. AT (LIT) is the set of all well formed atoms (literals) over $\mathfrak{V}$, $\mathfrak{C}$, $\mathfrak{F}$ and the alphabet of predicate symbols P.

For each literal L, $|L|$ is the atom of L and L^c denotes L's complement which is obtained from L by applying (or omiting) the negation sign not. The clause language C is the set of all clauses, a is the empty clause.

For a set D of terms, literals or clauses, vars (D) is defined as the set of all variable symbols in D. The subscript *gr* abbreviates ground, e . (T_{gr} i s the set of all ground terms .

When concerned with equality reasoning, we use E as the syntactic equality sign and assume $E \in P$. S^E denotes the extension of the clause set S by all functionally-reflexive axioms [WR73]. The set of all equality atoms AT^E is defined as $AT^E = \{E(q\ r) | q,r \in T\}$.

SUB is the set of all Substitutions and e denotes the Identity substitution. The domain o f a substitution σ, denoted $DOM(\sigma)$, i s the set of all variable symbols x with $\sigma x \neq x$. The codomaln of σ, denoted $COD(\sigma)$, is defined as $\sigma(DOM(\sigma))$. We say that two substitutions θ and λ *agree* on a subset V of $\mathfrak{V}$, denoted $\theta = \lambda[V]$, iff $\theta x = \lambda x$ for each $x \in V$.

A (ground) term rewriting system is a set of directed equations $R = \{E(q_i r_i) \in AT^E_{gr} | i \in J\}$ where $J \subset \mathbb{N}$. We define $\rightarrow_R$ by $q \rightarrow_R r$ iff $E(q\ r) \in R$ and we use $\rightarrow_R$ to denote the *reduction relation* associated with R (cf. [HO80]). $\xrightarrow{+}_R$ is the *transitive closure* of $\rightarrow_R$.

We shall also manipulate ground literals by a term rewriting system and extend $\rightarrow_R$ in the obvious way. For a set I of literals, the term rewriting system R(I) contained In I is defined as $R(I) = I \cap AT^E_{gr}$.

Given a variable disjoint set S of clauses and a clause C, $S \vdash C$ denotes the existence of a deduction of C from S, using the inference rules of factorization, resolution and paramodulation. $S \vdash_{\bar{R}} C$ is a deduction without paramodulation and $S \vdash_{\bar{P}} C$ is a deduction without resolution. S_{gr} denotes the set of all ground Instances of the clauses in S.

A (possibly infinite) subset I of LIT_{gr} is called an Interpretation iff for each LEI, $L^c \notin I$. I is E-closed iff for each $L \in I$ and each $K \in LIT_{gr}$, KGI whenever $L \rightarrow_{R(I)} K$. I i. an E-interpretation iff I is E-closed and $E(t\ t) \in I$ for each $t \in T_{gr}$

The *ΣRP-Calculus* Let $\mathfrak{S}$ be a set of sort symbols ordered by the subsort order $\leq_S$, i.e. a partial order which is reflexive, anti-symmetric and transitive. We use $s_1 \ll_s s_2$ as an abbreviation for: $s_1 <_s s_2$ such that ther is no s with $s_1 <_s s <_s s_2$. Variable, constant and function symbols are associated with a certain sort symbol, called the rangesort o the respective symbol. The sort [t] of a term t is the rangesort of the outermost function, constant or variable symbol of t.

All argument positions of a function or pre dicate symbol are associated with a certain sort symbol called the domalnsort. In the construction of the set T_Σ of all *Σ-terms* (or well sorted terms) and of the set AT_Σ (LIT_Σ) of all *Σ-atoms* (*Σ-literals*) of the ΣRP-calculus, only those terms may fill an argument position of a function or predicat symbol, whose sorts are subsorts of the domainsort given for the respective argument position. For equality atoms we assert $E(t_1 t_2) \in AT_\Sigma$ iff $t_1, t_2 \in T_\Sigma$.

A Σ-*clause* is a finite subset of LIT_Σ. The *many-sorted language* $\mathcal{L}_\Sigma$ is the set of all Σ-clauses. $T_{\Sigma gr}$ denotes the set of all variable free Σ-terms. $AT_{\Sigma gr}$, $LIT_{\Sigma gr}$ and $\mathcal{L}_{\Sigma gr}$ are defined in a similar way.

Sometimes we use sort symbols also as unary predicate symbols. Then we assume that s is the only domainsort of $s \in \mathcal{S}$ and we define $LIT^{\$}$ ($LIT^{\$}_\Sigma$) as the set of all ($\Sigma$-)literals in the *extended language* $\mathcal{L}^{\$}$ ($\mathcal{L}^{\$}_\Sigma$).

The set A^Σ of all *sort axioms* is the smallest variable disjoint subset of $\mathcal{L}^{\$}_\Sigma$ which satisfies

(1) $\{s(c)\} \in A^\Sigma$, if $c \in \mathcal{C}$ has rangesort s,

(2) $\{not\ s_1(x_1),\ldots,not\ s_k(x_k), s(f(x_1\ldots x_k))\} \in A^\Sigma$, if $f \in \mathcal{F}$ has $s_1,\ldots,s_k$ as domainsorts and has rangesort s, and each $x_i \in \mathcal{V}$ has rangesort s_i, and

(3) $\{not\ s_1(x), s_2(x)\} \in A^\Sigma$, if $s_1 \ll_s s_2$.

The *relativization* $\hat{C}$ of a Σ-clause C is a clause in $\mathcal{L}^{\$}_\Sigma$ and defined as
$\hat{C} = \{not\ s_1(x_1),\ldots,not\ s_n(x_n)\} \cup C$, where $\{x_1,\ldots,x_n\} = vars(C)$ and each x_i has rangesort s_i.
For a set S of Σ-clauses, $\hat{S}$ abbreviates $\{\hat{C} \in \mathcal{L}^{\$}_\Sigma \mid C \in S\}$.

A Σ-*substitution* σ is a substitution satisfying

(*) $\sigma x \in T_\Sigma$ and $[\sigma x] \le_s [x]$, for each $x \in \mathcal{V}$.

It can be shown that (*) is equivalent to $\sigma(T_\Sigma) \subset T_\Sigma$. SUB_Σ denotes the set of all Σ-substitutions. A Σ-*renaming substitution* ν for a set D of variables, literals or clauses is a renaming substitution for D such that $[\nu x] = [x]$ for each $x \in \mathcal{V}$.

For a Σ-clause C and a Σ-substitution σ, σC is called a Σ-*ground instance* of C, if $\sigma C \in \mathcal{L}_{\Sigma gr}$. A set D of Σ-terms or Σ-atoms is Σ-*unifiable* iff D is unifiable with a Σ-substitution σ. Then σ is a Σ-*unifier* of D. σ is a Σ-most general unifier (Σ-*mgu*) of D iff σ is mgu of D and $\sigma \in SUB_\Sigma$.

A Σ-substitution μ is a *weakening substitution* for a set $V \subset \mathcal{V}$ iff μ satisfies

(1) $COD(\mu) \subset \mathcal{V} \setminus V$,

(2) $\mu_{|V}$ is injective , and

(3) $[\mu x] <_s [x]$, if $x \in DOM(\mu)$.

For each $V \subset \mathcal{V}$, WSUB(V) denotes the set of all weakening substitutions for V. Obviously $\varepsilon \in WSUB(V)$ and $WSUB(V) \subset SUB_\Sigma$.

For a (ground) term rewriting system $\mathcal{R}$ we define the Σ-*reduction relation* $\rightarrow_{\Sigma\mathcal{R}}$ associated with $\mathcal{R}$ by $\rightarrow_{\Sigma\mathcal{R}} = \rightarrow_\mathcal{R} \cap (T_{\Sigma gr} \times T_{\Sigma gr})$. $\xrightarrow{+}_{\Sigma\mathcal{R}}$ is the transitive closure of $\rightarrow_{\Sigma\mathcal{R}}$. $\mathcal{R}$ is a Σ-*maximal* term rewriting system iff $\rightarrow_\mathcal{R} = \xrightarrow{+}_{\Sigma\mathcal{R}}$. Note that in general a Σ-maximal term rewriting system is *infinite*.

If a clause C is a resolvent, paramodulant or a factor of some Σ-clauses, then C is a Σ-*resolvent*, Σ-*paramodulant* or Σ-*factor* respectively, provided the mgu used to form C is a Σ-mgu and in the case of paramodulation, the *modulant literal* [Lov78] is a Σ-literal.

If $C \in \mathcal{L}_\Sigma$ and $\mu \in WSUB(V)$ for some $V \supset vars(C)$, then μC is a *weakened variant* of C. Obviously, each Σ-resolvent, Σ-paramodulant, Σ-factor and each weakened variant is a Σ-clause.

Given a variable disjoint set of Σ-clauses S, $S \vdash_\Sigma C$ denotes the existence of a Σ-deduction of C from S, i.e. $S \vdash C$ and each clause in this deduction is a Σ-renamed Σ-resolvent, Σ-paramodulant, Σ-factor or weakened variant. $S \vdash_{\Sigma R} C$ denotes a Σ-deduction without Σ-paramodulants and $S \vdash_{\Sigma P} C$ is a Σ-deduction without Σ-resolution.

Given a set of Σ-clauses S, $S_{\Sigma gr}$ denotes the set of all Σ-ground instances of the Σ-clauses in S. An interpretation I Σ-*satisfies a* Σ-*clause* C iff I satisfies each Σ-ground instance σC of C, i.e. $\sigma C \cap I \neq \emptyset$. I Σ-*satisfies a set of* Σ-*clauses* S iff I Σ-satisfies each clause in S. In this case, I is a Σ-*model* of S and S is Σ-*satisfiable*. If in addition I is an E-interpretation, then I ΣE-*satisfies* S, I is a ΣE-*model* of S and S is ΣE-*satisfiable*.

Throughout this paper $\langle \mathcal{S}, \leq_s \rangle$ is a partially ordered set of sort symbols and S stands for any variable disjoint set of Σ-clauses.

3. *The Weakening Rule*

The following examples should provide some motivation for our work illustrating the notions introduced so far and also demonstrate that the ΣRP-calculus is incomplete without the weakening rule. Essentially there are two reasons for this incompleteness:

(1) the Unification Theorem [Rob65] does not hold in the ΣRP-calculus,

(2) paramodulation is incomplete in the ΣRP-calculus (without the weakening rule).

Example 4.1 Let $\mathcal{S} = \{A,B,C,D\}$ with $D \ll_s B \ll_s A$ and $D \ll_s C \ll_s A$ and let $P \in \mathbf{P}$, $d \in \mathbf{C}$ and $\{u,v,w\} \subset \mathfrak{V}$ such that A is the only domainsort of P, D is the rangesort of d and w, B is the rangesort of u and C is the rangesort of v. Now consider the set of Σ-clauses $S = \{\{P(u)\},\{not\ P(v)\}\}$. S is Σ-unsatisfiable, because $S_{\Sigma gr} = \{\{P(d)\},\{not\ P(d)\}\}$ is unsatisfiable. But neither $\sigma = \{u \leftarrow v\}$ nor $\tau = \{v \leftarrow u\}$ are Σ-substitutions, i.e. no Σ-resolvent can be derived from the two clauses in S. But with the weakening rule, we find a Σ-refutation of S:

(C1)	$\forall u.$	$\{P(u)\}$	, given
(C2)	$\forall v.$	$\{not\ P(v)\}$	, given
(W1)	$\forall w.$	$\{P(w)\}$	,
(R2)		$\square$	.

W1 is a weakened variant μC1 of C1, where $\mu = \{u \leftarrow w\}$, and R2 is a Σ-resolvent of C2 and W1, because $\theta = \{v \leftarrow w\}$ is a Σ-mgu of $\{P(v), P(w)\}$.

Note that we propose the weakening rule as an additional inference rule only in order to isolate the crucial point and to obtain completeness results. In a *proof procedure* this rule is realized by a modification of the *unification algorithm*, i.e. in our system [BES81,Ohl82] the empty clause is derived from C1 and C2 by a single resolution step using the substitution $\theta \circ \mu = \{u \leftarrow w, v \leftarrow w\}$.

However this modification of the unification algorithm only covers applications of the weakening rule as in the above example. Unfortunately there are cases which cannot be solved using the modified unification algorithm:

Example 4.2 Let $\mathcal{S} = \{A,B\}$ with $B \ll_s A$ and let $P \in \mathbf{P}$, $\{b_1,b_2\} \subset \mathbf{C}$ and $\{x,y,z\} \subset \mathfrak{V}$ such that B is the only domainsort of P, A is the rangesort of x and y, and B is the rangesort of b_1, b_2 and z. $S = \{\{P(b_1)\}, \{E(x\ y)\}, \{not\ P(b_2)\}\}$ is a ΣE-unsatisfiable set of Σ-clauses because $S_{\Sigma gr} = \{\{P(b_1)\},\{E(b_1\ b_2)\},\ldots,\{not\ P(b_2)\}\}$ is E-unsatisfiable. We can derive four paramodulants from S, namely $\{P(x)\}$, $\{P(y)\}$, $\{not\ P(x)\}$ and $\{not\ P(y)\}$ none of which is a Σ-clause, i.e. not a Σ-paramodulant. But with the weakening rule we find a Σ-refutation of S:

(C1)		$\{P(b_1)\}$	, given
(C2)	$\forall x,y.$	$\{E(x\ y)\}$	, given
(C3)		$\{not\ P(b_2)\}$	, given
(W4)	$\forall x,z.$	$\{E(x\ z)\}$	,
(P5)	$\forall z.$	$\{P(z)\}$	,
(R6)		$\square$	.

W4 is a weakened variant μC2 of C2, where $\mu = \{y \leftarrow z\}$, P5 is a Σ-paramodulant of C1 and W4 and R6 is a Σ-resolvent of P5 and C3.

4. *Term Rewriting under Sorts*

We present a result for (ground) term rewriting under sorts, which is essential for the completeness of the ΣRP-calculus in the ground case:

> *Σ-Rewrite Theorem* If $\mathbf{R}$ is a Σ-maximal term rewriting system, then $\xrightarrow{+}_{\mathbf{R}} \cap (T_{\Sigma gr} \times T_{\Sigma gr}) = \xrightarrow{+}_{\Sigma\mathbf{R}}$.

The Σ-Rewrite Theorem obviously also holds for $\mathbf{R}$-rewrites of ground literals.

5. *Ground Completeness of the ΣRP-Calculus*

At the Σ-ground level there is no difference between resolution and Σ-resolution. Hence the main difficulty is in showing the result for paramodulation, i.e. to prove a result which links Σ-deductions $\vdash_{\Sigma P}$ to deductions $\vdash_P$ from a set of Σ-ground clauses. To this effect we need the following definition:

> *Definition 5.1*
> $Par(S) = \{C \in \mathcal{L}_{gr} \mid S^E_{\Sigma gr} \vdash_P C\}$,
> $Par_\Sigma(S) = \{C \in \mathcal{L}_{\Sigma gr} \mid S^E_{\Sigma gr} \vdash_{\Sigma P} C\}$, and
> $RPAR(S) = \{C \in \mathcal{L}_{gr} \mid S^E_{\Sigma gr} \vdash_P C$, such that no clause in $\vdash_P$ is obtained by paramodulating into a positive equality literal$\}$.

As a prerequisite for the proof of the Ground Completeness Theorem we show that if $Par_\Sigma(S)$ is satisfiable, then RPar(S) is satisfiable. As a first result we obtain:

> *Lemma 5.1* If $Par_\Sigma(S)$ is satisfiable, then it has a model $M_o \subset LIT_{\Sigma gr}$, such that $\mathcal{R}(M_o)$ is a Σ-maximal term rewriting system.

The above model M_o is E-closed, relative to $\xrightarrow{+}_{\Sigma\mathcal{R}(M_o)}$, i.e. we can prove the following:

> *Lemma 5.2* If $L \in M_o$ and $K \in LIT_{gr}$ such that $L \xrightarrow{+}_{\Sigma\mathcal{R}(M_o)} K$, then $K \in M_o$.

Now let the *rewrite-closure* M_o^* of M_o be M_o extended by all ground literals which are obtained by rewriting the ground literals in M_o (which are different from equality atoms) using the reduction relation $\xrightarrow{+}_{\mathcal{R}(M_o)}$.

Using the above lemmata, we can prove that the rewrite-closure is always an interpretation:

> *Lemma 5.3* If $Par_\Sigma(S)$ is satisfiable, then it has a model $M_o \subset LIT_{\Sigma gr}$, such that M_o^* is an interpretation.

If we assume by contradiction that M_o^* contains a pair of ground literals Q and Q^c, then we can find a pair of Σ-ground literals L and K in M_o such that $L \xrightarrow{+}_{\mathcal{R}(M_o)} K^c$. But $K^c \in LIT_{\Sigma gr}$ since $K \in LIT_{\Sigma gr}$, hence by the Σ-Rewrite Theorem $L \xrightarrow{+}_{\Sigma\mathcal{R}(M_o)} K^c$ because $\mathcal{R}(M_o)$ is Σ-maximal by Lemma 5.1. Using Lemma 5.2 we obtain $K^c \in M_o$, i.e. $\{K, K^c\} \subset M_o$ contradicting that M_o is an interpretation.

It is easily proved by induction on the length n of a deduction $S^E_{\Sigma gr} \vdash_P C$, that M_o^* satisfies each clause $C \in RPar(S)$, i.e. we can prove the following:

> *Lemma 5.4* If $Par_\Sigma(S)$ is satisfiable, then it has a model M_o such that M_o^* is a model of RPar(S).

and we finally obtain the

> *Ground Completeness Theorem for ΣRP* If $S_{\Sigma gr}$ is E-unsatisfiable, then $S^E_{\Sigma gr} \vdash_\Sigma \square$.

If $S_{\Sigma gr}$ is E-unsatisfiable, then Par(S) is E-unsatisfiable because $S_{\Sigma gr} \subset Par(S)$. By Theorem 1 from [WR73] we infer that Par(S) is unsatisfiable, hence RPar(S) is unsatisfiable [Lov78] and by contraposition of Lemma 5.4 $Par_\Sigma(S)$ is unsatisfiable. But then $S^E_{\Sigma gr} \vdash_\Sigma \square$ by the completeness of resolution [Rob65], because there is no difference between resolution and Σ-resolution in the Σ-ground case and each clause in $Par_\Sigma(S)$ can be obtained by a Σ-deduction $\vdash_{\Sigma P}$ from $S^E_{\Sigma gr}$.

6. *Unification under Sorts*

Since the Unification Theorem does not hold for unification under sorts, we must content ourselves with a weaker result. For unification under sorts, the notion of Σ-*compatibility* plays a central role:

> *Definition 6.1* Let $D \subset T_\Sigma$ be unifiable. D is Σ-*compatible* iff $[x] \le_s [y]$ or $[x] \ge_s [y]$ for all $x, y \in vars(D)$ with $\tau x = \tau y$, where τ is an mgu of D.

We can show that each Σ-unifiable set of Σ-compatible Σ-terms possesses a Σ-mgu:

> *Lemma 6.1* Let $D \subset T_\Sigma$ be Σ-unifiable. If D is Σ-compatible, then there exists a Σ-mgu of D.

This is proved by showing that among the mgu's of D (which are equivalent up to variable renamings), we can always find a Σ-mgu of D.

For arbitrary sets of Σ-unifiable Σ-terms, we enforce the existence of a Σ-mgu using a weakening substitution:

> *Σ-Unification Theorem* Given $D \subset T_\Sigma$, $V \subset \mathcal{V}$ and $\theta \in SUB_\Sigma$ with vars(D) $\subset$ V and θ unifies D, there exist $\mu, \sigma, \lambda \in SUB_\Sigma$ such that
> (1) $\mu \in WSUB(V)$,
> (2) σ is an mgu of μD , and
> (3) $\theta = \lambda \circ \sigma \circ \mu[V]$.

It is easily proved, that for each Σ-unifiable set of Σ-terms D, there exists some $\mu \in WSUB(V)$ such that μD is still Σ-unifiable and in addition is Σ-compatible. By Lemma 6.1 there exists a Σ-mgu σ of μD and since σ is an mgu we find some $\lambda \in SUB_\Sigma$ such that condition (3) is satisfied.

Note that the Σ-Unification Theorem obviously also holds for Σ-unifiable sets of Σ-atoms.

7. *Soundness and Completeness of the ΣRP-Calculus*

The soundness of the ΣRP-calculus is an immediate consequence of the following:

> *Soundness Lemma for ΣRP* Let M be an E-interpretation and $C \in \mathcal{L}_\Sigma$. If M Σ-satisfies S and $S \vdash_\Sigma C$, then M Σ-satisfies C.

The completeness theorem is shown as usual for the one-sorted calculus: We prove the Lifting Lemmata for Σ-Resolution and for Σ-Paramodulation in order to justify the Lifting Theorem for Σ-Deductions.

> *Lemma 7.1* (Lifting Lemma for Σ-Resolution) Given $A, B \in \mathcal{L}_\Sigma$, $L_A \in A$, $L_B \in B$ and $\theta \in SUB_\Sigma$ such that A and B share no variable symbols, L_A and L_B are complementary and θ unifies $\{|L_A|, |L_B|\}$, there exist a Σ-factor A* of a weakened variant of A, a Σ-factor B* of a weakened variant of B, a pair of complementary literals $L_A^* \in A^*$ and $L_B^* \in B^*$, weakened variants ρA^* and ρB^* and some $\lambda, \sigma \in SUB_\Sigma$ such that
> $$Res(\theta A, \theta L_A, \theta B, \theta L_B, \varepsilon) = \lambda\, Res(\rho A^*, \rho L_A^*, \rho B^*, \rho L_B^*, \sigma) .$$

The proof is similar to the proof of the Lifting Lemma in the RP-calculus. The existence of the various weakened variants is guaranteed by the Σ-Unification Theorem.

With an analogous *Lifting Lemma for Σ-Paramodulation*, we can prove the *Lifting Theorem for Σ-Deductions* as in the one-sorted calculus and using the Ground Completeness Theorem for ΣRP, we finally obtain the

> *Completeness Theorem for ΣRP* If S is ΣE-unsatisfiable, then $S^E \vdash_\Sigma \square$.

8. *The Sort-Theorem*

In this section the Sort-Theorem for the ΣRP-calculus is stated. We need the following two lemmata:

> *Lemma 8.1* If M is an E-model of $(\hat{S} \cup A^\Sigma)_{gr}$, then M E-satisfies $S_{\Sigma gr}$.

This lemma is used to prove one direction of the equivalence stated in the Sort-Theorem. The reverse direction is obtained with more difficulty:

> *Lemma 8.2* If $S_{\Sigma gr}$ is E-satisfiable, then $(\hat{S} \cup A^\Sigma)_{gr}$ is E-satisfiable.

We obtain an E-model M* of $(\hat{S} \cup A^\Sigma)_{gr}$ from an E-model $M \subset LIT_{gr}$ of $S_{\Sigma gr}$ by setting $M^* = (M \cup \lceil M \rceil) \cup (M \cup \lceil M \rceil)^c$, where the *kernel* $\lceil M \rceil$ of M is the set of all literals of form s(t), $t \in T_{gr}$ and $s \in \mathcal{S}$, for which there exists a $q \in T_{\Sigma gr}$ with $[q] \leq_{\mathcal{S}} s$ and $E(q\ t) \in M$, and the *$\mathcal{S}$-complement* $(M \cup \lceil M \rceil)^c$ of $(M \cup \lceil M \rceil)$ is the set of all literals of the form *not* s(t), $t \in T_{gr}$ and $s \in \mathcal{S}$, with $s(t) \notin (M \cup \lceil M \rceil)$. It can be proved that M* is an E-interpretation (whenever M is an E-interpretation with $M \cap LIT^{\mathcal{S}}_{gr} = \emptyset$), and that M* satisfies $(\hat{S} \cup A^\Sigma)_{gr}$.

The connection between the RP- and the ΣRP-calculus is now established by the

> *Sort-Theorem for the ΣRP-Calculus* S is ΣE-unsatisfiable iff $(\hat{S} \cup A^{\Sigma})$ is E-unsatisfiable.

S is ΣE-unsatisfiable iff $S_{\Sigma gr}$ is E-unsatisfiable, which is equivalent to the E-unsatisfiability of $(\hat{S} \cup A^{\Sigma})_{gr}$ by Lemmata 10.1 and 10.2. But $(\hat{S} \cup A^{\Sigma})_{gr}$ is E-unsatisfiable iff $(\hat{S} \cup A^{\Sigma})$ is E-unsatisfiable by the Herbrand Theorem and by the Compactness Theorem.

9. *An Automated Theorem Prover for the ΣRP-Calculus*

In [Wal82a] a survey is presented of how an automated theorem prover based on the RP-calculus can be modified to obtain an automated theorem prover for the ΣRP-calculus. The necessary modifications concern the *input-language* compiler, the *skolemization* routine, the *unification* algorithm, and the *computation* of factors, resolvents and paramodulants. The Markgraf Karl Refutation Procedure [BES81,Ohl82] was adapted to the ΣRP-calculus according to the modifications stated above. Below a proof protocol of the new system is presented, proving a many-sorted version of the well-known *monkey-banana-problem* [Lov78]:

```
**************************************************************************************
*                                                                                    *
*       MARKGRAF KARL REFUTATION PROCEDURE, UNI KARLSRUHE, VERSION  12-OCT-82        *
*                                                                                    *
*       DATE:   2-NOV-82 16:46:27                                                    *
*                                                                                    *
**************************************************************************************

AXIOMS GIVEN TO THE THEOREM PROVER:
     SORT ANIMAL,TALL:IN.ROOM
     TYPE BANANA,FLOOR:IN.ROOM
     TYPE CHAIR:TALL
     TYPE MONKEY:ANIMAL
     TYPE CAN.REACH(ANIMAL IN.ROOM)
     TYPE CLOSE.TO(IN.ROOM IN.ROOM)
     TYPE ON(IN.ROOM IN.ROOM)
     TYPE UNDER(IN.ROOM IN.ROOM)
     TYPE CAN.MOVE.NEAR(ANIMAL IN.ROOM IN.ROOM)
     TYPE CAN.CLIMB(ANIMAL TALL)
     AXM1 :  ALL X:ANIMAL Y:IN.ROOM NOT CLOSE.TO(X Y)  OR  CAN.REACH(X Y)
     AXM2 :  ALL X:ANIMAL Y:TALL NOT ON(X Y)  OR  NOT UNDER(Y BANANA)
                OR CLOSE.TO(X BANANA)
     AXM3 :  ALL X:ANIMAL Y:IN.ROOM Z:IN.ROOM NOT CAN.MOVE.NEAR(X Y Z)
                OR CLOSE.TO(Z FLOOR) OR UNDER(Y Z)
     AXM4 :  ALL X:ANIMAL Y:TALL NOT CAN.CLIMB(X Y)  OR  ON(X Y)
     AXM5 :  CAN.MOVE.NEAR(MONKEY CHAIR BANANA)
     AXM6 :  NOT CLOSE.TO(BANANA FLOOR)
     AXM7 :  CAN.CLIMB(MONKEY CHAIR)

THEOREM GIVEN TO THE THEOREM PROVER:
     THM8 :  NOT CAN.REACH(MONKEY BANANA)


AXM2 + AXM3 --> RES1 :  ALL X:ANIMAL Y:TALL Z:ANIMAL CLOSE.TO(X BANANA)
                           OR NOT ON(X Y) OR  CLOSE.TO(BANANA FLOOR)
                           OR NOT CAN.MOVE.NEAR(Z Y BANANA)
RES1 + AXM5 --> RES2 :  ALL X:ANIMAL CLOSE.TO(BANANA FLOOR)  OR  NOT ON(X CHAIR)
                           OR  CLOSE.TO(X BANANA)
AXM1 + RES2 --> RES3 :  ALL X:ANIMAL CAN.REACH(X BANANA)  OR  NOT ON(X CHAIR)
                           OR  CLOSE.TO(BANANA FLOOR)
AXM6 + RES3 --> RES4 :  ALL X:ANIMAL NOT ON(X CHAIR)  OR  CAN.REACH(X BANANA)
THM8 + RES4 --> RES5 :  NOT ON(MONKEY CHAIR)
RES5 + AXM4 --> RES6 :  NOT CAN.CLIMB(MONKEY CHAIR)
RES6 + AXM7 --> RES7 :  EMPTY

THE FOLLOWING CLAUSES WERE USED IN THE PROOF:
AXM7 AXM4 AXM5 AXM3 AXM2 RES1 RES2 AXM1 RES3 AXM6 RES4 THM8 RES5 RES6 RES7 .

THE THEOREM IS PROVED. END OF PROOF  2-NOV-82 16:47:22.
```

In our system, we use the expressions (cf. [Wal82b])

SORT $s_1,\ldots,s_n$:s to denote $s_i \ll_s s$,
TYPE $c_1,\ldots,c_n$:s to denote that $c_i \in \mathcal{C}$ has rangesort s ,
TYPE $P(s_1 \ldots s_n)$ to denote that $P \in \mathcal{P}$ has the domainsorts $s_1,\ldots,s_n$, and
ALL x:s to denote the universal quantification of a variable symbol x with rangesort s.

The system also solved the monkey-banana-problem, using the one-sorted axiomatization from [Lov78]. The following diagram shows the proof statistics of both examples runs:

CPU-SECONDS USED:	3.32	11.38	29 %
NUMBER OF STEPS EXECUTED:	7	16	44 %
NUMBER OF LINKS GENERATED:	22	99	22 %
NUMBER OF LINKS IN INITIAL GRAPH:	8	23	35 %
NUMBER OF CLAUSES GENERATED:	15	29	52 %
INITIAL CLAUSES:	8	13	
RESOLVENTS:	7	12	
FACTORS:	0	4	
NUMBER OF LITERALS GENERATED:	28	75	37 %
IN INITIAL CLAUSES:	14	24	
IN DEDUCED CLAUSES:	14	51	
LEVEL OF PROOF:	7	12	58 %
NUMBER OF CLAUSES IN PROOF:	15	25	60 %
G-PENETRANCE:	1.00	0.86	(#CLAUSES IN PROOF / #CLAUSES GENERATED)
D-PENETRANCE:	1.00	0.75	(#DEDUCED CLAUSES IN PROOF / #CLAUSES DEDUCED)

The first column lists the statistical values for the proof using the many-sorted calculus,the second column lists the values for the one-sorted calculus and the third column shows the ratio between the values of both example runs.

In the proof statistics, the value for 'number of links generated' corresponds to the size of the search space,the value for •number of steps executed' is a measure of the expense of the actual search and 'level of proof represents the search depth.

The comparison between the statistical values of both protocols immediately reveals the advantages of using an automated theorem prover based on the ZRP-calculus. The values are typical for all examples (and of course for more complex ones) that have been proved by this system.

Acknowledgement

I would like to thank my colleagues *K.-H. Blasius hi. eisinger.*,* H.J. *Ohlbach* and W. *Stephan* for their support during the preparation of this work, and also A. *Herold* for the useful discussions on Unification Theory. J. *Siekmann* read an earlier draft of this paper; his helpful criticism and support greatly contributed to the present form of this paper.

[BES81] Blasius, K., Eisinger.K, N., Siekmann, J. , Smolka, G. , Herold, A., and C. Waltker The Markgraf Karl Refutation Procedure. Proc. of the 7th International Joint Conference on Artificial Intelligence (1981)

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

[Cha78] Ckampe.aux, V. do. A Theorem Prover Dating a Semantic Network. Proc. of AISB/GI Conference, Hamburg (1978)

[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, University 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 theorie de la demonstration (These Paris), Warsaw (1930), chapter 3. Also in "Logical Writings" (W.D. Goldfarb ed.), D. Reidel Publishing Company (1971)

[H080] Huet, G., and V.C. Oppen Equations and Rewrite Rules: A Survey. Technical Report CSL-111, SRI International (1980)

[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)

[Lov78] Loveland, V.W. Automated Theorem Proving: A Logical Basis. North-Holland Publishing Company (1978)

[0be6 2] oberschelp A. Untersuchungen zur mehrsortigen Quantorenlogik. Mathematische Annalen 145 (1962)

[Ohl82] Oklback, H.J. The Markgraf Karl Refutation Procedure: The Logic Engine. Interner Bericht 24/82, Institut fur Informatik I, Universitat Karlsruhe (1982)

[Rob65] Robinbon, J.A. A Machine-Oriented Logic Based on the Resolution Principle. JACM 12 (1965)

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

[Sch51] Schmidt, A. Die Zulassigkeit der Behandlung mehrsortiger Theorien mittels der ublichen einsortigen Pradikatenlogik. Mathematische Annalen 123 (1951)

[Wal82a] Walther, C. A Many-Sorted Calculus Based on Resolution and Paramodulation. Interner Bericht 34/82, Institut fur Informatik I, Universitat Karlsruhe (1982)

[Wal82b] Walther, C. The Markgraf Karl Refutation Procedure: PLL - A First-Order Language for an Automated Theorem Prover. Interner Bericht 35/82, Institut fur Informatik I, Universitat Karlsruhe (1982)

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

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

[WR73] Wos, L. and G. Robinson Maximal Models and Refutation Completeness: Semidecision Procedures in Automatic Theorem Proving. In "Wordproblems" (W.W. Boone, F.B. Cannonito, R.C. Lyndon, eds.), North-Holland Publishing Company (1973)