Combining Induction Axioms By Machine

Christoph Walther
Technische Hochschule Darmstadt
Fachbereich Informatik, AlexanderstraBe 10
D 6100 Darmstadt
Germany

Abstract

The combination of induction axioms is investigated. It is shown how a pair of competing induction axioms (which e.g. are suggested by a heuristic of an induction theorem prover on a specific verification problem) are combined yielding a new induction axiom. The relation implicitly defined by the new axiom is the set-theoretic union of the well-founded relations implicitly defined by the induction axioms initially given. The proposed approach is non-heuristic but safe in the sense that an induction proof with the new axiom can be obtained whenever an induction proof with one of the given axioms would have been successful. Based on a result of *Bachmair* and *Dershowitz* for proving term rewriting systems noctherian, a commutation test is developed as a deductive requirement to verify the soundness of the combined axiom: It is shown how so-called commutation formulas can be derived by machine from the given axioms such that a verification of these formulas (e.g. by an induction theorem prover) guarantees the well-foundcdness of the relation defined by the combined axiom. Examples are presented to demonstrate the usefulness and strength of the proposed technique.

1 Introduction

One crucial point in proving theorems by induction is to find an induction axiom for a given conjecture ψ as the *right* instance of the *Generalized Principle of Noetherian Induction*: $[\forall m \in M.\ [\forall k \in M.\ k <_M m \rightarrow \psi(k)] \rightarrow \psi(m)] \rightarrow \forall m \in M.\ \psi(m)$. The invention of a "successful" well-founded relation $<_M$ for a statement constitutes the creativity of a human, and also of an automated expert in induction theorem proving. This problem has two aspects: (1) a *proof strategic* aspect, which means that the relation $<_M$ induced upon *allows to prove* the *induction formula* $[\forall m \in M.\ [\forall k \in M.\ k <_M m \rightarrow \psi(k)] \rightarrow \psi(m)]$, and (2) a *soundness* aspect, which means that the relation $<_M$ induced upon must be indeed *well-founded.*

To be successful we should use a relation $<_M$ which - considered as a *set* - is as large as possible because when proving $\psi(m)$, called the *induction conclusion*, we may use $\psi(k)$ for all k with $k <_M m$ as *induction hypotheses*. Therefore, the larger $<_M$ is, the more induction hypotheses $\psi(k)$ are available. As a consequence, if two well-founded relations $<_{M_1}$ and $<_{M_2}$ are to our disposal such that $<_{M_1} \subset <_{M_2}$, we always choose to induce upon $<_{M_2}$ because it is guaranteed that whenever we find a proof using $<_{M_1}$, we also find a proof using $<_{M_2}$. We call this the *subset principle*. But if neither $<_{M_1} \subset <_{M_2}$ nor $<_{M_2} \subset <_{M_1}$ is known, we may choose to induce upon $<_{M_1} \cup <_{M_2}$. By the subset principle this approach is *safe* in the sense that we always find a proof by induction upon $<_{M_1} \cup <_{M_2}$ whenever an induction upon $<_{M_1}$ or $<_{M_2}$ would have been successful. But it may happen that the relations $<_{M_1}$ and $<_{M_2}$ are too small to support a successful induction whereas the induction upon $<_{M_1} \cup <_{M_2}$ may work. It is shown in this paper that our proposal to combine induction axioms not only is safe, but is also useful and yields strong and successful induction axioms.

However, the union of well-founded relations is not necessarily well-founded, so well-foundedness of $<_{M_1} \cup <_{M_2}$ has to be verified to guarantee the soundness of the approach. We therefore demonstrate how formulas can be derived by machine, the truth of which entail the soundness of the combined induction axiom. Thus, new and useful induction axioms arc computed, leaving the verification of their soundness to the induction theorem proving system.

Our proposal to compute new induction axioms can be integrated into all induction theorem provers based on the *explicit induction* paradigm, e.g. the systems described in [Aubin, 1979; Boyer and Moore, 1979; Bundy et al., 1991; Biundo et al., 19861. It is currently being implemented in the INKA induction theorem prover, a system under development at the Technische Hochschule Darmstadt.

2 Computing Induction Axioms

The operation of an induction theorem proving system based on the explicit induction paradigm can be sketched in the following way (see [Walther, 1992] for a more detailed account): After some function is defined by the user of the system, a so-called *relation description* (*r-description* for short) is computed from the user's definition in a uniform way. Given, for instance, the functions

```
function quot(x,y:number):number ⇐
    if $y = 0$ then 0
    if $x < y$ then 0
    if $x \geq y \wedge y \neq 0$ then succ(quot(x−y y))
```

and

```
function half(x:number):number ⇐
    if $x = 0$ then 0
    if $x = 1$ then 0
    if $x \neq 0 \wedge x \neq 1$ then succ(half((x−1)−1))
```

as input, the r-descriptions $D_{quot} = \{(x \geq y \wedge y \neq 0, \{\{x/x-y, y/y\}\})\}$ and $D_{half} = \{(x \neq 0 \wedge x \neq 1, \{\{x/(x-1)-1\}\})\}$ are computed from the recursive cases of the algorithms.[1] An r-description D is a finite and non-empty set $\{C_1,...,C_k\}$ of *atomic* r-descriptions $C_i = (\varphi_i, \Delta_i)$, where the *range formula* φ_i is a quantifier-free formula and Δ_i is a finite and non-empty set of substitutions, called the *domain substitutions*. The set $rV(D)$ of *relevant variables* in D is the set of all variables mentioned in D, e.g. *x* and *y* in D_{quot}, and the set of *induction variables* $iV(D)$ of D contains all relevant variables which are substituted for by the domain substitutions. Each r-description D defines a *relation* $<_D$ on some cartesian product of the set of constructor ground terms, e.g. D_{quot} defines a relation $<_{quot} \subset (\mathcal{T}(\Sigma^c)_{number} \times \mathcal{T}(\Sigma^c)_{number})^2$ given as (a b) $<_{quot}$ (c d) iff $\mathfrak{I}[\![x/c, y/d]\!] \models x \geq y \wedge y \neq 0$, $a = \mathfrak{I}[\![x/c, y/d]\!](x-y)$ and $b = \mathfrak{I}[\![x/c, y/d]\!](y)$. $\mathfrak{I}$ is the *interpretation* for the known functions, which accepts some ground term from $\mathcal{T}(\Sigma)$ as input, e.g. plus(succ(0) succ(0)), and always returns a *constructor* ground term from $\mathcal{T}(\Sigma^c)$ as output, e.g. succ(succ(0)). Also $\mathfrak{I}(q)=q$ may be assumed for all constructor ground terms q. The interpretation $\mathfrak{I}$ can be applied also to terms with variables if *variable assignments* like $[\![x/c, y/d]\!]$ are used. For the above r-descriptions, (2,3) $<_{quot}$ (5,3) $<_{quot}$ (8,3) $<_{quot}$ ... is a chain wrt. the relation defined by D_{quot} and 1 $<_{half}$ 3 $<_{half}$ 5 $<_{half}$ ... is a chain wrt. the relation defined by D_{half}. The relations obtained are well-founded because only *terminating* functions are accepted by an induction theorem prover, and therefore these r-descriptions are called well-founded.

Since each well-founded r-description D defines a well-founded relation $<_D$ and likewise a well-founded set, we may uniformly associate an induction axiom with D. Given an r-description $D = \{(\varphi_1, \Delta_1),...,(\varphi_k, \Delta_k)\}$ with relevant variables x* and a formula ψ with free variables x*, D is associated with *k+1 induction formulas*, the *base case* ψ_0 and the *step formulas* $\psi_1,...,\psi_k$. The range formulas φ_i in the atomic r-descriptions of D define the cases of the induction steps and the domain substitutions $\delta \in \Delta_i$ are used to form the induction hypotheses. The base case is obtained as the complement of all the range formulas in D and $[\psi_0 \wedge ... \wedge \psi_k \rightarrow \forall x^*.\ \psi]$ is an *induction axiom*.

Consider, for instance, some formula $\psi[x,y]$ with free variables *x* and *y*. The induction formulas to prove $\forall x,y.\ \psi[x,y]$ are computed from D_{quot} as

$\psi_0 = \forall x,y.\ x < y \vee y=0 \rightarrow \psi[x,y]$ and

$\psi_1 = \forall x,y.\ x \geq y \wedge y \neq 0 \wedge \psi[x/x-y, y/y] \rightarrow \psi[x,y]$.

But if D_{half} is used instead of D_{quot}, the induction formulas for proving $\forall x,y.\ \psi[x,y]$ are obtained as

$\psi_0 = \forall x.\ x=0 \vee x=1 \rightarrow \forall y.\ \psi[x,y]$ and

$\psi_1 = \forall x.\ x \neq 0 \wedge x \neq 1 \wedge \forall y.\ \psi[x/(x-1)-1, y] \rightarrow \forall y.\psi[x,y]$.

The r-descriptions obtained from terminating functions necessitate further computations for subsequent induction proofs: Domain generalizations and range generalizations are computed and each range generalization then is separated which finally yields an r-description providing certain proof-technical advantages.[2] For instance, the well-founded r-description $D_{quot'} = \{(x \geq y \wedge y \neq 0, \{\{x/x-y\}\})\}$ is a *domain generalization* of the above r-description D_{quot} having only *x* as an induction variable. Domain generalizations are obtained by removing some substitution pairs from the domain substitutions thus extending the domain of the defined relation, as $D_{quot'}$ is obtained by removal of *y/y*. This extends the defined relation to (a b) $<_{quot'}$ (c d) iff $\mathfrak{I}[\![x/c, y/d]\!] \models x \geq y \wedge y \neq 0$ and $a = \mathfrak{I}[\![x/c, y/d]\!](x-y)$. Note that b may be *any* constructor ground term here because now no replacement is demanded for the relevant variable *y*. Consequently, (2,9) $<_{quot'}$ (5,3) $<_{quot'}$ (6,1) $<_{quot'}$ ... is a chain wrt. the relation defined by $D_{quot'}$. To prove $\forall x,y.\ \psi[x,y]$, the following induction formulas are obtained if the domain generalization $D_{quot'}$ is used:

$\psi_0 = \forall x,y.\ x < y \vee y=0 \rightarrow \psi[x,y]$ and

$\psi_1 = \forall x,y.\ x \geq y \wedge y \neq 0 \wedge \forall u.\ \psi[x/x-y, y/u] \rightarrow \psi[x,y]$.

The variable *y* in this example is *universally quantified* (after being renamed to *u*) in the induction hypothesis $\psi[x/x-y, y/u]$, and this yields a stronger induction than with D_{quot}.

The well-founded r-description $D_{quot''} = \{(x \neq 0 \wedge y \neq 0, \{\{x/x-y\}\})\}$ is a *range generalization* of the above r-description $D_{quot'}$. It is obtained by replacing the range formula in $D_{quot'}$ by some weaker requirement, thus extending the range of the defined relation. For instance, (2,9) is a minimal element wrt. $<_{quot'}$ because $2 \geq 9$ does not hold. But (2,9) is not minimal wrt. $<_{quot''}$ because $2 \neq 0 \wedge 9 \neq 0$, and therefore (0,...) $<_{quot''}$ (2,9). The advantage of a range generalization is that the fewer minimal elements a well-founded relation has, the fewer elements have to be treated as base cases of an induction based on this relation. To prove $\forall x,y.\ \psi[x,y]$, for instance, the following induction formulas are obtained if the range generalization $D_{quot''}$ is used:

$\psi_0 = \forall x,y.\ x=0 \vee y=0 \rightarrow \psi[x,y]$ and

$\psi_1 = \forall x,y.\ x \neq 0 \wedge y \neq 0 \wedge \forall u.\ \psi[x/x-y, y/u] \rightarrow \psi[x,y]$.

[1] To ease readability we often use the usual mathematical notation, e.g. $x>y$, $x+y$, $x-y$, $x-1$, 2 etc., instead of the terms formally required, e.g. gt(x y)=true, plus(x y), minus(x y), pred(x), succ(succ(0)) etc.

[2] The computation of a *domain generalization* corresponds to the computation of a *measured subset* as defined by [Boyer and Moore, 1979], the computation of a *range generalization* corresponds to the computation of a *revised machine* and the computation of a *separation* corresponds to the computation of a *superimposed machine*.

Using its termination proof facility, cf. [Boyer and Moore, 1979; Walther, 1988], an induction theorem prover checks each domain and range generalization for well-foundedness and those which fail the test are not considered any further. After this test only the generalized (well-founded) r-descriptions are used for computing induction axioms. This does not restrict provability, because $<_D \subset <_{D'}$ if D' is an r-description obtained from D by a domain or a range generalization.

If some statement ψ has to be verified by the induction theorem prover, r-descriptions are *suggested* by an *induetion heuristic* in order to compute induction axioms from them, cf. [Aubin, 1979; Boyer and Moore, 1979]. By recognizing function "calls" in the statement, the r-descriptions associated with the called functions arc selected from the set of well-founded r-descriptions already computed. However, those r-descriptions which fail to meet certain variable requirements are disregarded. The remaining r-descriptions are modified by replacing some relevant non-induction variables with the corresponding terms in the call (if necessary).

Given, for instance, the statement $\forall x.\ \psi_1 \equiv \forall x.\ \text{quot}(x\ 4) = \text{half}(\text{half}(x))$, the r-descriptions $D_{\text{quot}''} = \{(x{\neq}0 \wedge 4{\neq}0, \{\{x/x{-}4\}\})\}$ and $D_{\text{half}'} = \{(x{\neq}0, \{\{x/(x{-}1){-}1\}\})\}$ are suggested by an induction heuristic. The r-description $D_{\text{half}'}$ is a range generalization of D_{half} and $D_{\text{quot}''}$ is built from the r-description $D_{\text{quot}'}$ by replacing the non-induction variable *y* of $D_{\text{quot}''}$ with 4 to match the argument in the call.

3 Combining Induction Axioms

The r-descriptions obtained as suggestions of an induction heuristic are rated and compared, e.g. by using the *subsumption heuristic* [Boyer and Moore, 1979] or the *containment test* [Walther, 1992], and those which fail the tests are rejected. But in general more than one r-description survives the comparison, e.g. $D_{\text{quot}''}$ and $D_{\text{half}'}$ in the example above, and we have to decide what to do with them. Obviously, statement ψ_1 can be easily proven by induction upon $<_{\text{quot}''}$ (but not upon $<_{\text{half}'}$). However, this is only known after a proof is computed. So, the problem is to find a well-founded relation which *guarantees* provability whenever an induction proof using either the $<_{\text{quot}''}$-or the $<_{\text{half}'}$-induction is successful. We simply use $<_{\text{quot}''} \cup <_{\text{half}'}$ and with the subset principle provability is not destroyed. So the induction axiom for ψ_1 is computed as:

$$\forall x.\ x{=}0 \rightarrow \text{quot}(x\ 4){=}\text{half}(\text{half}(x))\ ,$$
$$\forall x.\ x{\neq}0 \wedge \text{quot}(((x{-}1){-}1)\ 4){=}\text{half}(\text{half}((x{-}1){-}1))$$
$$\wedge\ \text{quot}(x{-}4\ 4){=}\text{half}(\text{half}(x{-}4)) \rightarrow \text{quot}(x\ 4) = \text{half}(\text{half}(x))$$
$$\overline{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}$$
$$\forall x.\ \text{quot}(x\ 4) = \text{half}(\text{half}(x))\ .^3$$

The induction hypothesis $\text{quot}(((x{-}1){-}1)\ 4) = \text{half}(\text{half}((x{-}1){-}1))$ stems from the $<_{\text{half}'}$-induction and the induction hypothesis $\text{quot}(x{-}4\ 4) = \text{half}(\text{half}(x{-}4))$ stems from the $<_{\text{quot}''}$-induction. Both induction formulas can be easily proven. Therefore statement ψ_1 is an *inductive truth* which intuitively means that all quantifiers in a formula range over constructor ground terms only. We define $\mathfrak{I} \models \forall x.\varphi$ *iff* $\mathfrak{I}[x/q] \models \varphi$ for *all* constructor ground terms q which may be substituted for x and $\text{Th}_{\text{Ind}} := \{\psi \mid \psi$ is a closed formula such that $\mathfrak{I} \models \psi\}$. An induction theorem prover tests whether a formula ψ is a member of the set Th_{Ind} of all inductive true theorems.[4]

A straightforward idea for formalizing this approach is to use the union $D_1 \cup D_2$ of a pair of r-descriptions D_1 and D_2 as the r-description from which the induction axiom is computed, because $<_{D_1 \cup D_2} = <_{D_1} \cup <_{D_2}$. But generally *non valid* induction formulas are generated if the union of r-descriptions is used. In particular $D_{\text{quot}''} \cup D_{\text{half}'} = \{(x{\neq}0, \{\{x/x{-}4\}\}), (x{\neq}0, \{\{x/(x{-}1){-}1\}\})\}$ yields two step formulas for statement ψ_1, viz.

(1) $\forall x.\ x{\neq}0 \wedge \text{quot}(x{-}4\ 4){=}\text{half}(\text{half}(x{-}4))$
$\rightarrow \text{quot}(x\ 4) = \text{half}(\text{half}(x))$, and

(2) $\forall x.\ x{\neq}0 \wedge \text{quot}(((x{-}1){-}1)\ 4) = \text{half}(\text{half}((x{-}1){-}1))$
$\rightarrow \text{quot}(x\ 4) = \text{half}(\text{half}(x))$,

one of which is not provable by first-order means: The induction hypothesis can be used to prove step formula (1), but this is not the case for step formula (2). The reason for this failure is that not all required induction hypotheses wrt. $<_{\text{quot}''} \cup <_{\text{half}'}$ are provided by step formula (2). However, this problem can be avoided if a *separated* r-description is used. This is an r-description D such that the range formulas of D exclude each other, i.e. $[\forall x^*\ \neg\varphi_1 \vee \neg\varphi_2] \in \text{Th}_{\text{Ind}}$ for all members (φ_1, Δ_1) and (φ_2, Δ_2) of D, and a separated r-description guarantees that all possible induction hypotheses wrt. $<_D$ are provided by a step formula, cf. [Walther, 1992]. To avoid unseparated r-descriptions when forming the union of relations, we propose the *separated union*, i.e. a separated r-description such that the relation implicitly defined is the union of the relations implicitly defined by the constituting r-descriptions:

Definition 3.1 For a pair of r-descriptions $D_1 = \{(\varphi_1, \Delta_1),\ldots,(\varphi_k, \Delta_k)\}$ and $D_2 = \{(\eta_1, \Theta_1),\ldots,(\eta_h, \Theta_h)\}$, the *separated union* $D_1 \oplus D_2$ of D_1 and D_2 is the smallest r-description such that for each atomic r-description $(\varphi_i, \Delta_i) \in D_1$ and each atomic r-description $(\eta_j, \Theta_j) \in D_2$

(1) $(\varphi_i \wedge \eta_0, \Delta_i) \in D_1 \oplus D_2$,
(2) $(\varphi_0 \wedge \eta_j, \Theta_j) \in D_1 \oplus D_2$, and
(3) $(\varphi_i \wedge \eta_j, \Delta_i \cup \Theta_j) \in D_1 \oplus D_2$,

where φ_0 abbreviates $\neg\varphi_1 \wedge \ldots \wedge \neg\varphi_k$ and η_0 abbreviates $\neg\eta_1 \wedge \ldots \wedge \neg\eta_h$.

Theorem 3.1 [Walther, 1991] For each pair of separated r-descriptions D_1 and D_2: (1) $D_1 \oplus D_2$ is a separated r-description, and (2) $<_{D_1 \oplus D_2} = <_{D_1} \cup <_{D_2}$.

[3] Here and subsequently obvious simplifications are used, like $4{\neq}0$ can be omitted here. We also abbreviate "$\forall$x:number" by "$\forall$x.".

[4] An induction theorem prover only provides a *sufficient* requirement for $\psi \in$ Thind because, due to *Godel's First Incompleteness Theorem,* neither Thind is decidable nor semi-decidable.

Since $<_{D_1} \subset <_{D_1 \oplus D_2}$ and $<_{D_2} \subset <_{D_1 \oplus D_2}$ by Theorem 3.1, we may always use the separated union $D_1 \oplus D_2$ instead of D_1 or D_2 to compute an induction axiom. With the subset principle it is guaranteed that an induction proof with $D_1 \oplus D_2$ always can be obtained whenever an induction proof with D_1 or with D_2 is successful. We compute $D_{quot"} \oplus D_{half} = \{(x \neq 0, \{\{x/(x-1)-1\}, \{x/x-4\}\})\}$, and the above induction axiom for ψ_1 was obtained from this r-description.

4 Commuting Induction Axioms

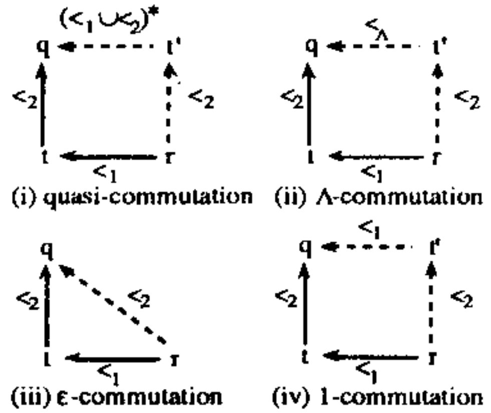
However, since the union of well-founded relations generally is not well-founded, the well-foundedness of $D_1 \oplus D_2$ has to be verified before it is used to compute an induction axiom. For solving this problem the termination proof facility of an induction theorem prover may be used. Actually this will be our ultimate resort, but first we shall attempt to verify the well-foundedness of $D_1 \oplus D_2$ directly from the well-foundedness of D_1 and D_2.

To do so we borrow a technique from the area of *term rewriting.* There a problem quite similar to ours arises, viz. to prove that a term rewriting system $R_1 \cup R_2$ is noetherian, where R_1 and R_2 are noetherian term rewriting systems. For this problem the notion of *quasi-commutation* has been proposed:

Definition 4.1 [Bachmair and Dershowitz, 1986] For a pair of relations $<_1$ and $<_2$ on some set S, $<_1$ *quasi-commutes over* $<_2$ iff (*) for all $q,t,r \in S$ with $q <_2 t <_1 r$ some $t' \in S$ exists such that $q <^*_{1\cup 2} t' <_2 r$, cf. Figure 1(i), where $<^*_{1\cup 2}$ denotes the reflexive and transitive closure of $<_1 \cup <_2$.

Theorem 4.1 [Bachmair and Dershowitz, 1986] If $<_1$ quasi-commutes over $<_2$, then $<_1 \cup <_2$ is well-founded iff $<_1$ and $<_2$ are well-founded.

Quasi-commutation can be used to verify the well-foundedness of $<_1 \cup <_2$ as demonstrated by Theorem 4.1: If $<_1$ and $<_2$ are well-founded, only "$<_1$ quasi-commutes over $<_2$" or else "$<_2$ quasi-commutes over $<_1$" has to be verified for proving the well-foundedness of $<_1 \cup <_2$.[5]



(i) quasi-commutation (ii) Λ-commutation

(iii) ε-commutation (iv) 1-commutation

Figure 1

To check for quasi-commutation in our application, we have to formulate a syntactical requirement in terms of relation descriptions which is sufficient for requirement (*) in Definition 4.1. This is necessary because our relations are not explicitly given but are implicitly defined as the semantics of r-descriptions. So requirement (*) in Definition 4.1 essentially is a *semantical* property which necessitates a *syntactical* counterpart for our purposes. But requirement (*) cannot be characterized by a finite set of formulas because the transitive closure $<^+$ cannot be characterized. So we have to be satisfied with those approximations of requirement (*) which are finitely representable and we define:

Definition 4.2 For a pair of relations $<_1$ and $<_2$ on some set S, $<_1$ quasi-commutes over $<_2$ *with type* Λ iff (**) for all $q,t,r \in S$ with $q <_2 t <_1 r$ some $t' \in S$ exists such that $q <_\Lambda t' <_2 r$, where Λ is some *finite* string over {1,2}, $a <_{\Lambda \bullet d} b$ stands for "some $c \in S$ exists such that $a <_\Lambda c <_d b$", and $a <_\varepsilon b$ means $a = b$ (where ε is the empty string).

Figure 1(ii) illustrates Definition 4.2. Obviously, quasi-commutation with type Λ entails quasi-commutation, but not vice versa because Λ is a *fixed* string. For instance, $<_1$ quasi-commutes over $<_2$ with *type* ε iff for all $q,t,r \in S$ with $q <_2 t <_1 r$ also $q <_2 r$, cf. Figure 1(iii), and $<_1$ quasi-commutes over $<_2$ with *type 1* iff for all $q,t,r \in S$ with $q <_2 t <_1 r$ some $t' \in S$ exists such that $q <_1 t' <_2 r$, cf. Figure 1(iv).

We have eliminated transitivity by using quasi-commutation with a *type* and are able to represent this weaker requirement in terms of relation descriptions and formulas derived from them. For the sake of brevity we shall confine ourselves here to the cases of quasi-commutation with type ε and type 1. Moreover, we shall confine ourselves to relation descriptions consisting of *one* atomic r-description with *one* domain substitution only, and refer to [Walther, 1991] for the general form. We start with the *type 1* case:

Definition 4.3 For a pair of r-descriptions $D_1 = \{(\varphi, \{\delta\})\}$ and $D_2 = \{(\eta, \{\theta\})\}$, D_1 *quasi-commutes over* D_2 with *type 1* (*1-commutes* for short) iff for some variables x^* :

(1) $rV(D_1) = Dom(\delta) = rV(D_2) = Dom(\theta) = x^*$,
(2) $[\forall x^*. \varphi \wedge \delta(\eta) \rightarrow \eta \wedge \theta(\varphi)] \in Th_{Ind}$, and
(3) $[\forall x^*. \varphi \wedge \delta(\eta) \rightarrow \bigwedge_{x \in x^*} \theta(\delta(x)) = \delta(\theta(x))] \in Th_{Ind}$.

Theorem 4.2 [Walther, 1991] If an r-description D_1 1-commutes over an r-description D_2, then $<_{D_1}$ quasi-commutes over $<_{D_2}$ with type 1.

Requirement (1) in Definition 4.3 is the *1-variable requirement* and the formulas in requirements (2) and (3) are the *1-commutation formulas*. If these formulas hold, then $<_{D_1}$ quasi-commutes over $<_{D_2}$ by Theorem 4.2, hence $<_{D_1} \cup <_{D_2}$ is well-founded by Theorem 4.1 (provided $<_{D_1}$ and $<_{D_2}$ are well-founded), and since $<_{D_1 \oplus D_2} = <_{D_1} \cup <_{D_2}$ by Theorem 3.1, $D_1 \oplus D_2$ is a well-founded r-description.

[5] We have *two* attempts to establish the well-foundedness of $<_1 \cup <_2$, because "quasi-commutes over" is not a symmetric property.

Hence we demand that an induction theorem prover verifies the 1-commutation formulas before $D_1 \oplus D_2$ is used to compute an induction axiom. For the above example, the 1-commutation formulas to verify that $D_{half'}$ 1-commutes over $D_{quot'''}$ are computed as:

(2) $\forall x.\ x \neq 0 \wedge ((x-1)-1) \neq 0 \rightarrow x \neq 0 \wedge (x-4) \neq 0$ and

(3) $\forall x.\ x \neq 0 \wedge ((x-1)-1) \neq 0 \rightarrow ((x-4)-1)-1 = ((x-1)-1)-4$.

Obviously formula (2) does not hold, so 1-commutation of $D_{quot'''}$ over $D_{half'}$ is tested:[6]

(2') $\forall x.\ x \neq 0 \wedge (x-4) \neq 0 \rightarrow x \neq 0 \wedge ((x-1)-1) \neq 0$ and

(3') $\forall x.\ x \neq 0 \wedge (x-4) \neq 0 \rightarrow ((x-1)-1)-4 = ((x-4)-1)-1$.

Also obviously these 1-commutation formulas are true and consequently the well-foundedness of $D_{quot'''} \oplus D_{half'}$ is verified. This proves the soundness of the induction axiom initially used for statement ψ_1, cf. Section 3.

Next we consider quasi-commutation of *type* ε:

Definition 4.4 For a pair of r-descriptions $D_1 = \{(\varphi, \{\delta\})\}$ and $D_2 = \{(\eta, \{\theta\})\}$, D_1 *quasi-commutes over* D_2 with *type* ε (*ε-commutes* for short) iff for $x^* = rV(D_1) \cup rV(D_2)$:

(1) all variables in $\eta \subset Dom(\delta)$,

(2) $[\forall x^*.\ \varphi \wedge \delta(\eta) \rightarrow \eta] \in Th_{Ind}$, and

(3) $[\forall x^*.\ \varphi \wedge \delta(\eta) \rightarrow \bigwedge_{x \in Dom(\theta)} \theta(\delta(x)) = \theta(x)] \in Th_{Ind}$.

Theorem 4.3 [Walther, 1991] If an r-description D_1 ε-commutes over an r-description D_2, then $<_{D_1}$ quasi-commutes over $<_{D_2}$ with type ε.

Requirement (1) in Definition 4.4 is the ε-*variable requirement*. The formulas in requirements (2) and (3) are the ε-*commutation formulas* and we demand that an induction theorem prover verifies these formulas before $D_1 \oplus D_2$ is used to compute an induction axiom.

As an example demonstrating the usefulness of ε-commutation, consider the functions minus1 and minus2 given as

```
function minus1(x,y:number):number ⇐
  if x≤y then 0
  if x>y then succ(minus1(x succ(y)))

function minus2(x,y:number):number ⇐
  if x=0 then 0
  if x≠0 ∧ y=0 then x
  if x≠0 ∧ y≠0 then minus2(x−1 y−1)
```

and the statement $\forall x,y.\ \psi_2 \equiv \forall x,y.$ minus1(x y) ≡ minus2(x y). An induction heuristic could suggest $D_{minus2'} = \{(x \neq 0, \{\{x/x-1\}\})\}$ and $D_{minus2''} = \{(y \neq 0, \{\{y/y-1\}\})\}$ both obtained by domain and range generalization from $D_{minus2} = \{(x \neq 0 \wedge y \neq 0, \{\{x/x-1,\ y/y-1\}\})\}$, and also suggest $D_{minus1} = \{(x>y, \{\{x/x, y/succ(y)\}\})\}$.

The tests for 1- and for ε-commutation fail for almost all combinations of the three r-descriptions, simply because the variable requirements are not satisfied. But the ε-variable requirement for verifying that D_{minus1} ε-commutes over $D_{minus2'}$ is satisfied and the ε-commutation formulas are computed as

(2) $\forall x,y.\ x>y \wedge x \neq 0 \rightarrow x \neq 0$ and

(3) $\forall x,y.\ x>y \wedge x \neq 0 \rightarrow (x-1)=(x-1)$.

The verification of both formulas is trivial. Therefore, $D_{minus1} \oplus D_{minus2'} = \{(x \neq 0 \wedge x \leq y,\ \{\{x/x-1\}\}),\ (x>y, \{\{x/x-1\}, \{x/x, y/succ(y)\}\})\}$ is a well-founded r-description from which the following induction axiom is computed for ψ_2:

$$\begin{array}{l}
\forall x,y.\ x=0 \qquad \rightarrow minus1(x\ y)=minus2(x\ y)\ , \\
\forall x,y.\ x \neq 0 \wedge x \leq y \wedge \forall z.\ minus1(x-1\ z)=minus2(x-1\ z) \\
\qquad \rightarrow minus1(x\ y)=minus2(x\ y)\ , \\
\forall x,y.\ x>y\ \wedge \forall z.\ minus1(x-1\ z)=minus2(x-1\ z) \\
\qquad \wedge\ minus1(x\ succ(y))=minus2(x\ succ(y)) \\
\qquad \rightarrow minus1(x\ y)=minus2(x\ y) \\
\hline
\forall x,y.\ minus1(x\ y)=minus2(x\ y)\quad .
\end{array}$$

The induction hypothesis minus1(x succ(y)) ≡ minus2(x succ(y)) stems from the $<_{minus1}$-induction and the induction hypothesis ∀z. minus1(x−1 z) ≡ minus2(x−1 z) (appearing in both step formulas) stems from the $<_{minus2'}$-induction. Since the variable z in the latter induction hypothesis is universally quantified, the induction is so strong that the statement can be proven with one trivial lemma only, viz. $[\forall x,y.\ x \neq 0 \wedge x \leq y \rightarrow y \neq 0]$ is required for the first step formula. Both induction hypotheses are used in the proof of the second step formula. The hypothesis stemming from $D_{minus2'}$ must be used *twice*, viz. with z substituted by y and with z substituted by y−1. The success of the $D_{minus1} \oplus D_{minus2'}$-induction is remarkable because the functions minus1 and minus2 *differ significantly* in their *recursion structure*. This usually necessitates that an induction theorem prover is supported by *additional lemmata* which relate one function to the other if only one of the suggested r-descriptions is used to form an induction axiom. This is not needed here, and we consider this as an evidence for the strength of the proposed method.

As a further example, consider the function plus given as

```
function plus(x,y:number):number ⇐
  if x=0 then y
  if x≠0 then succ(plus(x−1 y))
```

and the statement $\forall x,y.\ \psi_3 \equiv \forall x,y.$ plus(x y) ≡ plus(y x). An induction heuristic could suggest $D_{plus'} = \{(x \neq 0, \{\{x/x-1\}\})\}$ and also $D_{plus''} = \{(y \neq 0, \{\{y/y-1\}\})\}$, both obtained by a domain generalization from $D_{plus} = \{(x \neq 0, \{\{x/x-1, y/y\}\})\}$ (after a variable renaming for $D_{plus''}$ to match the argument in the call). All tests for 1- and for ε-commutation fail because the variable requirements are not satisfied. Analysing the case, we see that $D_{plus'} \oplus D_{plus''} = \{(x \neq 0 \wedge y=0, \{\{x/x-1\}\}), (x=0 \wedge y \neq 0, \{\{y/y-1\}\}), (x \neq 0 \wedge y \neq 0, \{\{x/x-1\}, \{y/y-1\}\})\}$ cannot be well-founded because

[6] The falsification of those formulas usually is trivial, at least if some facility looking for counterexamples for system-computed conjectures is applied, cf. [Protzen, 1992], yielding here x=3 as a counterexample.

$$\ldots <_{plus''} (0\ 1) <_{plus'} (1\ \ 0) <_{plus''} (0\ 1) <_{plus'} (1\ \ 0)$$

is an infinite $<_{plus' \cup plus''}$-chain in $\mathcal{T}(\Sigma^c)_{number} \times \mathcal{T}(\Sigma^c)_{number}$. But since $<_{plus'}$ and $<_{plus''}$ are well-founded, an obvious remedy is to use some *subset* $<_{plus*}$ of $<_{plus'}$ (or of $<_{plus''}$ respectively), hoping that $<_{plus* \cup plus''}$ is well-founded at least.

But fortunately we know some subset of $<_{plus'}$: Since $D_{plus'}$ was obtained from D_{plus} by a domain generalization, $<_{plus} \subset <_{plus'}$ holds (and therefore D_{plus} was ignored in favour of $D_{plus'}$ when r-descriptions were suggested for ψ_3). We now test D_{plus} and $D_{plus''}$ for quasi-commutation. Here the 1-variable requirement is not satisfied but the ε-variable requirement is when ε-commutation of D_{plus} over $D_{plus''}$ is tested. The ε-commutation formulas are computed as:

(2) $\forall x,y.\ x \neq 0 \wedge y \neq 0 \rightarrow y \neq 0$ and

(3) $\forall x,y.\ x \neq 0 \wedge y \neq 0 \rightarrow (y-1) = (y-1)$.

The verification of both formulas is trivial. Therefore, $D_{plus} \oplus D_{plus''} = \{(x \neq 0 \wedge y=0,\ \{\{x/x-1,\ y/y\}\}),\ (x=0 \wedge y \neq 0,\ \{\{y/y-1\}\}),\ (x \neq 0 \wedge y \neq 0,\ \{\{x/x-1,\ y/y\},\ \{y/y-1\}\})\}$ is a well-founded r-description from which the induction axiom for ψ_3 is computed as:

$$\begin{array}{ll}
\forall x,y.\ x=0 \wedge y=0 & \rightarrow plus(x\ y)=plus(y\ x)\ , \\
\forall x,y.\ x \neq 0 \wedge y=0 \wedge plus(x-1\ y)=plus(y\ x-1) & \\
 & \rightarrow plus(x\ y)=plus(y\ x)\ , \\
\forall x,y.\ x=0 \wedge y \neq 0 \wedge \forall z.\ plus(z\ y-1)=plus(y-1\ z) & \\
 & \rightarrow plus(x\ y)=plus(y\ x)\ , \\
\forall x,y.\ x \neq 0 \wedge y \neq 0 \wedge plus(x-1\ y)=plus(y\ x-1) & \\
\qquad \wedge \forall z.\ plus(z\ y-1)=plus(y-1\ z) & \\
 & \rightarrow plus(x\ y)=plus(y\ x)\ , \\
\hline
\multicolumn{2}{c}{\forall x,y.\ plus(x\ y)=plus(y\ x)\quad .}
\end{array}$$

The induction hypotheses $plus(x-1\ y) = plus(y\ x-1)$ stem from the $<_{plus}$-induction and the induction hypotheses $\forall z.\ plus(z\ y-1) = plus(y-1\ z)$ stem from the $<_{plus''}$-induction. Since the variable z in the latter induction hypotheses is universally quantified, the induction is so strong that the statement can be proven *without* an additional lemma. All induction hypotheses are required for the proofs of the step formulas, where the latter hypothesis must be used *twice* in the proof of the third step formula, viz. with z substituted by x, and with z substituted by x−1. Without using the relation description $D_{plus} \oplus D_{plus''}$ to form an induction axiom, *additional lemmata* such as $[\forall x.\ plus(x\ 0)=x]$ (to prove the base case) and $[\forall x,y.\ plus(x\ succ(y)) = succ(plus(x\ y))]$ (to prove the step case) are needed to verify statement ψ_3, cf. [Boyer and Moore, 1979]. This is not required here and we consider this as an evidence for the strength of the proposed method.

Based on the above analysis we demand for an implementation that an induction theorem prover considers the r-descriptions from which the domain generalizations were computed, if the quasi-commutation tests fail for the generalized forms, as we did for the example above.

5 Concluding Remarks

There are cases where commutation with a *type* may not help because quasi-commutation with a type is not a necessary requirement for quasi-commutation, which in turn is not a necessary requirement for the well-foundedness of the union of well-founded relations. If all commutation tests fail, then the termination proof facility of an induction theorem prover has to be used, cf. [Boyer and Moore, 1979; Walther, 1988], hoping that it can prove the well-foundedness of the separated union. However, this should always be the ultimate resort. Such direct well-foundcdness proofs usually are much more difficult than the proofs necessitated by the commutation tests, as it easily can be observed from the examples presented above. This is because the latter deduction problems embody the knowledge that the given relations are well-founded and, therefore, quite often can be proven by propositional reasoning and case analysis only.

References

[Aubin, 1979] R. Aubin. Mechanizing Structural Induction. *Theoretical Computer Science* 9:329-362, 1979.

[Bachmair and Dershowitz, 1986] L. Bachmair and N. Dershowitz, Commutation, Transformation, and Termination. In *Proc. 8th Intern. Conf. on Automated Deduction,* Oxford, 1986. Springer LNCS, vol. 230.

[Biundo *et aL,* 1986] S. Biundo, B. Hummel, D. Hutter, and C. Walther. The Karlsruhe Induction Theorem Proving System. In *Proc. 8th Intern. Conf. on Automated Deduction,* Oxford, 1986. Springer LNCS, vol. 230.

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

[Bundy *et al.,* 1991] A. Bundy, F. van Harmelen, J. Hesketh and A. Smaill. Experiments with Proof Plans for Induction. *J. Automated Reasoning,* 7(3):303-324, 1991.

[Protzen, 1992] M. Protzen. Disproving Conjectures. In *Proc. Ilth Intern. Conf. on Automated Deduction,* Saratoga Springs, 1992. Springer LNAI, vol 607.

[Walther, 1988] C. Walthcr. Argument-Bounded Algorithms as a Basis for Automated Termination Proofs. In *Proc. 9th Intern. Conf. on Automated Deduction,* Argonne, 1988. Springer LNCS, vol. 310. (revised version to appear in *Artificial Intelligence)*

[Walther, 1991] C. Walther. Computing Induction Axioms - Methods and Formal Bases. Research Memo, Fachbereich Informatik, Technische Hochschule Darmstadt, 1991.

[Walther, 1992] C. Walther. Computing Induction Axioms. In: *Proc. Conference on Logic Programming and Automated Reasoning,* St. Petersburg, 1992. Springer LNAI, vol 624.