Abstract Notions and Inference Systems for Proofs by Mathematical Induction

Claus-Peter Wirth* and Klaus Becker

Fb. Informatik, Universität Kaiserslautern, D-67663, Germany
{wirth, klbecker}@informatik.uni-kl.de

Abstract. Soundness of inference systems for inductive proofs is sometimes shown ad hoc and a posteriori, lacking modularization and interface notions. As a consequence, these soundness proofs tend to be clumsy, difficult to understand and maintain, and error prone with difficult to localize errors. Furthermore, common properties of the inference rules are often hidden, and the comparison with similar systems is difficult. To overcome these problems we propose to develop soundness proofs systematically by presenting an abstract frame inference system a priori and then to design each concrete inference rule locally as a sub-rule of some frame inference rule and to show its soundness by a small local proof establishing this sub-rule relationship. We present a frame inference system and two approaches to show its soundness, discuss an alternative, and briefly classify the literature. In an appendix we give an example and briefly discuss failure recognition and refutational completeness.

1 Motivation

Given some set of first-order axioms 'R', one is often not only interested in those properties 'Γ' which are logical consequences of 'R', i. e. which hold in all models of 'R': " $\mathrm{R} \models \Gamma$ "; but also in properties which are only required to hold in some specific sub-class of the class of models of 'R'. Instead of restricting the class of models by some required property, one may also ask for those 'Γ' for which (instead of " $\mathrm{R} \models \Gamma$ ") only " $\mathrm{R} \models \Gamma\tau$ " must hold for all τ taken from a specific set of (e. g. ground) substitutions. Notions of validity resulting from combinations of possible restrictions of these two kinds are usually called *inductive validity* and the inductively valid properties are called *inductive theorems*. In Wirth & Gramlich (1994) we discussed the most important of these notions in a unified framework on the basis of positive/negative-conditional equational specifications as introduced in Wirth & Gramlich (1993). These theorems are called "inductive" since (finite) proofs for most of them require mathematical induction. Inductive reasoning extends deductive reasoning by capturing infinite deductive proofs in a finite cyclic representation, e. g. capturing

* supported by the Deutsche Forschungsgemeinschaft, SFB 314 (D4-Projekt)

$$\dfrac{\Gamma(x_0)}{\dfrac{\Gamma(0)}{} \quad \dfrac{\Gamma(\mathsf{s}(x_1))}{\dfrac{\Gamma(\mathsf{s}(0))}{} \quad \dfrac{\Gamma(\mathsf{s}(\mathsf{s}(x_2)))}{\dfrac{\Gamma(\mathsf{s}(\mathsf{s}(0)))}{} \ \ldots}}} \qquad \text{(using “}x_i = 0 \ \vee\ \exists x_{i+1}\!: \ x_i = \mathsf{s}(x_{i+1})\text{”) in something like} \qquad \dfrac{\Gamma(x_0)}{\dfrac{\Gamma(0)}{} \quad \dfrac{\Gamma(\mathsf{s}(x_1))}{\dfrac{\Gamma(x_1)}{\text{(back to top)}}}}$$

(where the formulas below each line imply the formula above). For this kind of cyclic reasoning to be sound, the deductive reasoning must terminate for each instantiation of the theorem. This can be guaranteed when one requires for each cyclic reasoning the preconditions (usually called *induction hypotheses*) (e. g. “$\Gamma(x_1)$”) to be smaller than the *“induction” conclusion* (e. g. “$\Gamma(\mathsf{s}(x_1))$”) w. r. t. some wellfounded ordering, called *induction ordering*.

In Walther (1994) we can read the following about proving inductive theorems: “Research on automated induction these days is based on two competing paradigms: *Implicit induction* (also termed *inductive completion*, *inductionless induction*, or, less confusingly, *proof by consistency*) evolved from the *Knuth-Bendix Completion Procedure* .... The other research paradigm ... is called *explicit induction* and resembles the more familiar idea of induction theorem proving using induction axioms.”

While the two paradigms are not uniformly defined in the research community, we call the latter paradigm “explicit” because in the underlying inference systems each cyclic reasoning is made explicit in a single inference step which brings together induction hypotheses and conclusions in a set of *induction base* and *induction step* formulas and explicitly guarantees the termination of their cycles with the help of a sub-proof or -mechanism for the wellfoundedness of the induction ordering resulting from the step formulas.

The inference systems for implicit induction, however, permit us to spread the cyclic reasoning as well as the termination control over several inference steps. To rediscover the inductive cycles in the reasoning we usually have to inspect several inference steps instead of a single one that explicitly does the induction. Possibly since this seemed to be somewhat difficult compared to the older and well-known explicit induction, the paradox “inductionless induction” became a name for implicit induction. Another reason for this name might be the emphasis the researchers in the field of implicit induction gave to the refutational completeness (cf. Appendix B) of their inference systems: In general the set of inductively valid theorems is not enumerable for all reasonable and interesting notions of inductive validity; therefore refutational completeness is highly appreciated for an inference system for inductive theorem proving as it is an optimal theoretical quality. Refutational completeness, however, does not help to find finite proofs for inductively valid formulas (whereas the ability of an inductive theorem prover to detect invalid formulas is most important under a practical aspect (cf. Appendix A), especially for *inductive* theorem proving because of the important role generalizations play in it, where an invalid formula can result from a valid input theorem due to over-generalization).

To succeed in proving an inductive theorem in finite time, implicit inductive theorem provers have to solve exactly the same problem as explicit inductive theorem provers, namely to find a finite cyclic representation for an infinite deductive proof as well as an induction ordering guaranteeing the termination of its cycles. Therefore, if a theorem prover with sufficient deductive power fails to show an inductive theorem, then either it fails to construct the proper reasoning cycles or its mechanisms for bookkeeping of ordering information (cf. sect. 2) or for satisfying ordering constraints are too weak. While inference systems for implicit induction usually have sufficient potentiality to construct the proper reasoning cycles, their bookkeeping of relevant ordering information may be insufficient for certain proofs (cf. Appendix C for an example), even though their powerful orderings for satisfying the ordering constraints partially compensate for this insufficiency. Inference systems for explicit induction on the other hand do not require any bookkeeping of ordering information since the ordering information is only of local importance within single explicit induction steps. Explicit inductive theorem provers usually use rather simple semantic orderings which have turned out to be powerful enough for almost all practical applications. These provers, however, usually do not find more sophistically structured reasoning cycles, e. g. in mutually recursive domains or in the case (which may be of more importance in practice) that the required instantiations of the induction hypotheses are difficult to be guessed when the step formulas are synthesized and do not become obvious before the induction hypotheses can be applied to the induction conclusions (cf. Protzen (1994) for a simple example). From an abstract point of view and beyond the technicalities usually involved, however, all inductive proofs of our intuition can be formulated according to each of the two paradigms without changing the reasoning cycles.

Note, however, that the scope of mathematical induction (i. e. reasoning in terminating cycles) goes beyond inductive theorem proving (w. r. t. our definition above). A field of application of mathematical induction which is very similar to inductive theorem proving w. r. t. data and problem structures is to prove (ground) confluence for first-order clauses, cf. Becker (1993) and Becker (1994).

Therefore, in our opinion, while arguing for the one and against the other paradigm should be overcome, a unifying representation of all these induction-based approaches is not only theoretically possible, but also strongly required from a practical point of view: The possible combination of insight, methods, techniques, and heuristics based on results of research in all these fields will be beneficial for the search for proofs by mathematical induction in practice.

The abstract notions and the frame inference system we present in this paper are proposed as a top level step towards such a unification. This frame inference system is necessarily more similar to inference systems for implicit than for explicit induction, since, from a top level view and according to our definition of the terms, explicit induction is a form of implicit induction where the induction is restricted to be done explicitly within single inference steps. The results of the field of explicit induction, however, are most important for developing concrete inference systems consisting of practically useful sub-rules of our frame inference rules. While the construction of such concrete inference systems as well as a comparison and a combination of implicit and explicit induction are supported by the abstract framework of this paper, a proper treatment of these subjects cannot be given here but must be elaborated in several future papers.

2 Prover States

Prover states are intended to represent the state of the prover, i. e. to record which sub-tasks of a proof have been successfully established, which goals remain to be proved, etc.. Technically, the prover states are the field of our inference relation '$\vdash$'. For *deductive* reasoning we may start with a set 'G' of *goals* which contains the theorems we want to prove, transform them and finally delete them after they have become trivial tautologies. Such an inference relation, starting from the theorems to be proved and reducing them until some termination criterion is satisfied, is called *analytic*, cf. e. g. Bibel & Eder (1993). These theorems must belong to some set 'Form' which contains the formulas the inference system can treat. If '$\vdash$' permits only sound transformation and deletion steps, then from "$G \vdash^{\oplus} \emptyset$" (where '$\vdash^{\oplus}$' denotes the reflexive and transitive closure of '$\vdash$') we may conclude that 'G' is valid. For practical reasons, we would like to have a set 'L' of *lemmas* at hand. Into this 'L' inference steps can store axioms of the specification, already proved lemmas or the results of any theorem prover called for solving special tasks which might be helpful in our inference process. We can then use 'L' for transformations of 'G' that are only known to be sound relative to 'L'. Thus our prover states should be pairs of sets '(L, G)' such that "$(\emptyset, G) \vdash^{\oplus} (L, \emptyset)$" implies validity of '$G$' and '$L$'. For *inductive* reasoning, we additionally need a set 'H' of *induction hypotheses*. Similar to 'L', the set 'H' may be built up during the inference process and used for transformations of 'G' which are *founded* on 'H' in the sense that they are only known to be sound relative to 'H'. Similar to the treatment of lemmas, our prover states should be triples of sets '(L, H, G)' such that "$(\emptyset, \emptyset, G) \vdash^{\oplus} (L, H, \emptyset)$" implies validity of '$G$', '$L$', and '$H$'. Unlike the lemmas in 'L', however, the hypotheses in 'H' are not known to be valid before the whole induction proof is done (i. e. "$G = \emptyset$"). Here we are running the risk of being caught in a cyclic kind of reasoning like: "The goal can be deleted, since it is valid due to the hypothesis, which is valid, if the goal can be deleted, ...". On the other hand, some kind of cyclic reasoning is really required for successful inductive proofs of finite length. We only have to make sure that this cyclic reasoning terminates. This can be achieved by equipping each formula in 'H' or 'G' with a weight and allowing a hypothesis to transform a goal only if the weight of the hypothesis is smaller than the weight of the goal w. r. t. a *wellfounded quasi-ordering* '$\lesssim$' (i. e. some reflexive and transitive relation $\lesssim$, whose *ordering* $< \; := \; \lesssim \setminus \gtrsim$ does not allow infinite descending sequences $\daleth_0 > \daleth_1 > \daleth_2 > \ldots$), which we will call the *induction ordering* in what follows.

We would like to point out that we distinguish between the weight of a formula and the actual formula itself: For explicit induction, weights are not needed on the inference system level because each inductive reasoning cycle is encapsuled in a single inference step which combines induction conclusions with induction hypotheses into step formulas. For implicit induction, however, induction conclusions and hypotheses are not joined from the beginning. Instead, the conclusions are taken as goals and transformed until it becomes obvious which hypotheses will be useful for proving them; an idea which is just now coming into view of the researchers in the field of explicit induction, cf. Protzen (1994). At this point, when hypotheses are to be applied to the transformed goals, their weights are needed to transfer ordering information from the original goals that generated the hypotheses to the transformed goals. Roughly speaking, the goals must store the weights of hypotheses for which they carry the proof work. (This still permits mutual induction.) A possible weight for a formula, which is so natural that there are hardly any other weights in the literature on implicit induction, is (the value of a measure applied to) the formula itself. However, if we require the weight of a goal to be determined by the formula alone, then the chance to transform or delete this goal by means of some fixed hypothesis (which must be smaller w. r. t. '$<$') gets smaller with each transformation of the goal into other goals which are smaller w. r. t. '$<$'. Such transformation of goals is usually called *simplification*. While simplification of a goal is an important[2] heuristic, the weight of the goal should not change during simplification. This can be stated more generally: For concrete inference rules it is very important that a goal can transmit its weight unchanged to the goals which it is transformed into. This would be generally impossible if the weight of a formula were restricted to be the formula itself. In our approach, therefore, each element '$\beth$' of "$H \cup G$" is some *syntactic construct* from a set 'SynCons'. Besides its formula 'form($\beth$)', '$\beth$' may have some additional contents describing its weight. Theoretically, one can consider each syntactic construct to be a pair made up of a formula expressing some property and a weight carrying the ordering information for avoiding non-terminating cycles in the use of inductive arguments. For the description of concrete inference systems within our abstract framework, however, it is more convenient not to restrict the syntactic constructs to this form because in some of these inference systems the formulas share some structure with the weights: E. g., in Bachmair (1988) the formulas are the weights, and in Becker (1994) the weights restrict the semantics of the formulas by the so-called "reference compatibility". The distinction between formulas and syntactic constructs (i. e. formulas augmented with weights) has the following advantages compared to other inference systems for implicit induction:

Easier Design of Inference Rules: The design of concrete inference rules (e. g. as sub-rules of the abstract inference rules of sect. 5 below) becomes simpler because a transformation of the actual formula does not necessarily include an appropriate transformation of its weight (into a smaller one) and is thus not restricted by ordering constraints. For an illustrating example cf. Appendix C.

No Global Ordering Restriction: The design of inference steps becomes possible which transform the formula of a goal into another one which is bigger w. r. t. the induction ordering, cf. Gramlich (1989).

High Quality of Ordering Information: The loss of ordering information during simplification of a goal (as described above) is avoided, which (as far as we know) was first described in Wirth (1991), exemplified by a failure

2 when the induction ordering contains the evaluation ordering of the functional definitions of the specification, cf. Walther (1994)

of a formal induction proof just caused by this loss of ordering information. There it is also sketched how to store the weight of the goal to avoid this information loss, — an idea which is also to be found in Becker (1993). For an illustrating example cf. Appendix C.

Focus on Relevant Ordering Information: Some induction proofs are only possible if we do not measure the whole formula (as often is the case when a clause is measured as the multi-set of *all* its literals) but only some subformula, subterm or variable of it. Focusing on certain variables is common for human beings (speaking of "induction on variable x", e. g.). While for the mechanisms usually applied inside the induction rule of explicit induction focusing on certain variables (called "*measured variables*" in Boyer & Moore (1979) and Walther (1994)) is standard, for focusing in implicit induction a marking concept was introduced in Wirth (1991) due to practical necessity, exhibited by an example. The more general focusing that can be achieved with the syntactic constructs here, permits us not to measure those parts of formulas which (due to unsatisfiable ordering constraints) block the application of useful hypotheses, thereby permitting us to focus on the literals (or even terms, variables) that do get smaller on their way from the induction conclusion to the induction hypothesis. This permits additional applications of induction hypotheses. For an illustrating example cf. Appendix C.

All in all, the inference relation '$\vdash$' should operate on prover states which are triples "(L, H, G)" of finite sets such that 'L' contains formulas (from 'Form') and 'H' and 'G' contain syntactic constructs (from 'SynCons') whose formulas may be accessed via the function "form: SynCons $\rightarrow$ Form". While prover states being trees of syntactic constructs may be more useful in practice, the simpler data structure presented here suffices for the purposes of this paper.

3 Counterexamples and Validity

For powerful inductive reasoning we must be able to restrict the test for the weight of a hypothesis to be smaller than the weight of a goal (which must be satisfied for the permission to apply the hypothesis to the goal) to the special case semantically described by their formulas. This can be achieved by considering only such instances of their weights that result from ground substitutions describing invalid instances of their formulas. A syntactic construct augmented with such a substitution providing extra information on the invalidity of its formula is called a *counterexample*. A syntactic construct whose formula is valid thus has no counterexamples. We assume the existence of some set 'Info' describing this extra information and require the induction ordering '$\precsim$' to be a wellfounded quasi-ordering not simply on "SynCons" but actually on "SynCons$\times$Info". Furthermore, we require "being a counterexample" to be a well-defined basic property which must be either true or false for each $(\daleth, J) \in$ SynCons$\times$Info. Finally, in order to formally express the relation between counterexamples and our abstract notion of validity, we require for each syntactic construct $\daleth \in$ SynCons that 'form($\daleth$)' is valid iff there is no $J \in$ Info such that $(\daleth, J)$ is a counterexample.

Let us consider this final requirement for two instantiations of our abstract notion of validity of formulas:

For the case of inductive validity of a formula 'Γ' given as indicated in sect. 1 (i. e. iff $\models_{\mathcal{A}} \Gamma\tau$ for all "inductive substitutions" τ and all algebras $\mathcal{A}$ belonging to the class 'K' of those models which satisfy some additional property required for this kind of inductive validity) an appropriate way of satisfying the requirement is to define a formula to be a clause, to define a syntactic construct to be a pair (Γ, f) of a formula Γ and a weight f (described in terms of the variables of the formula), to define " $\mathrm{form}((\Gamma, f)) := \Gamma$ ", to define the elements of 'Info' to be triples "$(\tau, \mathcal{A}, \kappa)$" with $\mathcal{A} \in \mathrm{K}$ and κ valuating the remaining free variables of " $\Gamma\tau$ " to elements of the universe (or carrier) of the algebra $\mathcal{A}$, and then to say that " $((\Gamma, f), (\tau, \mathcal{A}, \kappa)) \in \mathrm{SynCons} \times \mathrm{Info}$ is a counterexample " if τ is an inductive substitution and $\mathcal{A}_\kappa$ evaluates " $\Gamma\tau$ " to false.

For the case of ground joinability (as defined in Becker (1993) or Becker (1994)), an appropriate way is to define a formula to be either a clause C (where validity means joinability of all ground instances $C\tau$) or a pair of clauses (C, D) (where validity means joinability of all those ground instances $C\tau$ for which $((C, D), \tau)$ is reference compatible), to define syntactic constructs to be pairs of clauses, 'form' to be the identity function, 'Info' to be the set of substitutions, and then to say that " $((C, D), \tau) \in \mathrm{SynCons} \times \mathrm{Info}$ is a counterexample " if τ is ground, $((C, D), \tau)$ is reference compatible, and " $C\tau$ " is not joinable.

Note that our notion of "counterexample" is a semantic one contrary to the notion of "inconsistency proof" used in Bachmair (1988). Generally speaking, an abstract frame inference system that is to be fixed prior to the design of concrete inference rules has to be sufficiently stable and therefore its notions should not rely on our changeable ideas on formal proofs.

Finally note that even with our emphasis here on proving valid formulas positively (instead of being refutationally complete), the somewhat negative kind of argumentation with counterexamples is handier, somewhat less operationally restricted, and more convenient for defining and proving properties of practically useful inference systems than the less local *formal proofs* used in the positive proving approach of Gramlich (1989) or Reddy (1990).

4 Foundedness

In this section we move from counterexamples to an even higher level of abstraction which allows the reader to forget about ' $\precsim$ ', 'Info', and counterexamples from sect. 5 on. We use the notion of counterexamples to lift ' $\precsim$ ' from "$\mathrm{SynCons} \times \mathrm{Info}$" to subsets of 'SynCons' by explaining what we mean by saying that a set H of hypotheses is *founded* on a set G of goals (written $H \curvearrowright G$) or by saying that a set G of goals is *strictly founded* on a set H of hypotheses (written $G \searrow H$ or $H \swarrow G$). Roughly speaking, $H \curvearrowright G$ indicates that the hypotheses are known to be valid if a *final* prover state (i. e. one with an empty set of goals) can be entailed. $H \swarrow G$ indicates that the goals in G can be deleted by the application of smaller hypotheses from H.

Definition 1 Foundedness. Let $M,H,G \subseteq \mathrm{SynCons}$. Let ' $\searrow\!/\!\curvearrowright$' be a symbol for a single relation. Now M is said to be *strict/quasi-founded* on (H,G) (denoted by $M \searrow\!/\!\curvearrowright (H,G)$) if
$\forall\,\aleph \in M\colon\ \forall I \in \mathrm{Info}\colon$

$$\left(\begin{array}{l}((\aleph,I) \text{ is a counterexample}) \quad \Rightarrow\\ \left(\begin{array}{l}\qquad\exists\beth \in H\colon\ \exists J \in \mathrm{Info}\colon \left(\begin{array}{l}\quad((\beth,J) \text{ is a counterexample})\\ \wedge\,(\aleph,I) > (\beth,J)\end{array}\right)\\ \vee\ \exists\beth \in G\colon\ \exists J \in \mathrm{Info}\colon \left(\begin{array}{l}\quad((\beth,J) \text{ is a counterexample})\\ \wedge\,(\aleph,I) \gtrsim (\beth,J)\end{array}\right)\end{array}\right)\end{array}\right).$$

M is said to be *strictly founded* on H (denoted by $M{\searrow}H$) if $M \searrow\!/\!\curvearrowright (H,\emptyset)$.
M is said to be (quasi-) *founded* on G (denoted by $M{\curvearrowright}G$) if $M \searrow\!/\!\curvearrowright (\emptyset,G)$.
Note that (for $\aleph \in \mathrm{SynCons}$) the expressive power of " $\{\aleph\} \searrow\!/\!\curvearrowright \ldots$ " is higher than that of " $\{\aleph\}{\searrow}\ldots$ " and " $\{\aleph\}{\curvearrowright}\ldots$ " together, since " $\{\aleph\}{\searrow}H \vee \{\aleph\}{\curvearrowright}G$ " implies " $\{\aleph\} \searrow\!/\!\curvearrowright (H,G)$ ", but the converse does not hold in general.

Corollary 2. *Let* $H \subseteq \mathrm{SynCons}$*. Now each of the following seven properties is logically equivalent to validity of* $\mathrm{form}[H]$*:*

(1) $H{\curvearrowright}\emptyset$	(4) $\forall G{\subseteq}\mathrm{SynCons}\colon\ H{\curvearrowright}G$
(2) $H{\searrow}\emptyset$	(5) $\forall G{\subseteq}\mathrm{SynCons}\colon\ H{\searrow}G$
(3) $H{\searrow}H$ *(due to wellfoundedness of '$>$')*	(6) $\exists G{\subseteq}\mathrm{SynCons}\colon\ ((\mathrm{form}[G]$ *is valid* $)\ \wedge\ H{\curvearrowright}G)$
	(7) $\exists G{\subseteq}\mathrm{SynCons}\colon\ ((\mathrm{form}[G]$ *is valid* $)\ \wedge\ H{\searrow}G)$

Corollary 3. *Let* $H \subseteq G \subseteq \mathrm{SynCons}$. *Now:* $\emptyset{\searrow}H{\curvearrowright}G$.

Corollary 4. *The following inclusion-properties hold:* $\searrow\ \subseteq\ \curvearrowright$. $\searrow\!\circ\!\curvearrowright\ \subseteq\ \searrow$. $\curvearrowright\!\circ\!\searrow\ \subseteq\ \searrow$.

Corollary 5.

(1)	$M_1 \curvearrowright N_1\ \wedge\ M_2 \curvearrowright N_2$	$\Rightarrow$	$M_1{\cup}M_2 \curvearrowright N_1{\cup}N_2$
(2)	$M_1 \searrow N_1\ \wedge\ M_2 \searrow N_2$	$\Rightarrow$	$M_1{\cup}M_2 \searrow N_1{\cup}N_2$
(3)	$M_1{\cup}M_2 \curvearrowright N_1$	$\Rightarrow$	$M_1 \curvearrowright N_1{\cup}N_2$
(4)	$M_1{\cup}M_2 \searrow N_1$	$\Rightarrow$	$M_1 \searrow N_1{\cup}N_2$
(5)	$G \searrow H$	$\Rightarrow$	$G \searrow H{\setminus}G$

Note that the last item of the previous as well as the first item of the following corollary rely on the wellfoundedness of '$>$'.

Corollary 6.

(1)	$M \searrow\!/\!\curvearrowright (H,G)\ \wedge\ H \curvearrowright G{\cup}M$	$\Rightarrow$	$M \curvearrowright G$
(2)	$M \searrow\!/\!\curvearrowright (H,G)\ \wedge\ G{\searrow}N$	$\Rightarrow$	$M \searrow H{\cup}N$

Corollary 7. $\curvearrowright$ *is a quasi-ordering.*

Corollary 8.
$\searrow$ *is a transitive relation, which is neither irreflexive nor generally reflexive.*
Let $\forall i{\in}\mathbb{N}\colon\ H_i,\ G_i \subseteq \mathrm{SynCons}$. *Then (by the wellfoundedness of* $>$*)* $\forall i{\in}\mathbb{N}\colon\ H_i{\searrow}H_{i+1}$ *implies that* $\mathrm{form}[H_i]$ *must be valid. More generally,* $\forall i{\in}\mathbb{N}\colon\ H_i \searrow\!/\!\curvearrowright (H_{i+1},G_{i+1})$ *implies* $H_i \curvearrowright \bigcup_{j>i} G_j$. *Moreover, the restriction of* $\searrow$ *to those* $H \subseteq \mathrm{SynCons}$ *with invalid* $\mathrm{form}[H]$ *is a wellfounded ordering.*

5 The Frame Inference System

We now come to four abstract inference rules defining '$\vdash$'. Thus, in this and the following three sections, '$\vdash$' will be restricted to application of one of the four following inference rules.

In what follows, let $\Gamma \in$ Form ; $\aleph, \gimel \in$ SynCons ; L, L' be finite subsets of 'Form'; and H, H', G, G', and M be finite subsets of 'SynCons':

Expansion:
$$\frac{(L\ ,\ H\ ,\ G\ )}{(L\ ,\ H\ ,\ G \cup \{\gimel\}\ )}$$

Hypothesizing:
$$\frac{(L\ ,\ H\ ,\ G\ )}{(L\ ,\ H \cup \{\aleph\}\ ,\ G\ )}$$
if L is invalid or $\{\aleph\} \curvearrowright H \cup G$

Acquisition:
$$\frac{(L\ ,\ H\ ,\ G\ )}{(L \cup \{\Gamma\}\ ,\ H\ ,\ G\ )}$$
if L is invalid or Γ is valid.

Deletion:
$$\frac{(L\ ,\ H\ ,\ G \cup \{\gimel\}\ )}{(L\ ,\ H\ ,\ G\ )}$$
if L is invalid or $\{\gimel\} \searrow\!/\!\!\curvearrowright (H, G)$

The Expansion rule has two typical applications. The first introduces sub-goals for a goal that is to be deleted, cf. the Transformation rule below. The second is the very difficult task of introducing new conjectures that are needed for the whole induction proof to work. The Hypothesizing rule makes a new hypothesis '$\aleph$' available for the proof. Since forward reasoning on hypotheses is hardly required, it can usually be restricted to the following sub-rule (cf. Corollary 3) which just stores the goals. This storing is necessary indeed because these goals usually have been transformed when they become useful for inductive reasoning.

Memorizing:
$$\frac{(L\ ,\ H\ ,\ G \cup \{\aleph\}\ )}{(L\ ,\ H \cup \{\aleph\}\ ,\ G \cup \{\aleph\}\ )}$$

The Acquisition rule makes a new lemma 'L' available for the proof. The rule may be used to include axioms from the specification or formulas proved in other successful runs of the inference system or by any other sound prover which seems appropriate for some special purposes. The Deletion rule permits the deletion of a goal that is strictly founded on some hypotheses. While we cannot to go into details on how to find this out, the Deletion rule especially permits us to remove a goal if its formula is implied by the formula of an instance of a hypothesis and this instance is smaller than the goal in our induction ordering. More frequently, however, is the Deletion rule used in the following combination with several preceding Expansion steps:

Transformation: $\dfrac{(L\ , H\ , G \cup \{\ \daleth\ \}\ )}{(L\ , H\ , G \cup M\quad)}$

if L is invalid or $\{\daleth\} \searrow/\!\curvearrowright (H, G{\cup}M)$

The Transformation rule replaces a goal '$\daleth$' with a (possibly empty) set 'M' of sub-goals whose completeness may rely on hypotheses from 'H' or lemmas from 'L'. It is the real working rule of the frame inference system. The intended design of concrete inference systems for specific kinds of validity mainly consists of finding corresponding sub-rules of the Transformation rule.

In the following sections we will present two alternative approaches for explaining why the above inference system implements the ideas presented in the beginning of sect. 2. The first is called "analytic" because it is based on an invariance property that holds for an initial state and is kept invariant by the analytic inference steps. The second is called "backwards" because it is based solely on an invariance property which holds for a final (i. e. successful) state and is kept invariant when one applies the inference rules in backward direction.

6 The Analytic Approach

The analytic approach was first formalized in Wirth (1991). In sect. 2 we indicated that if '$\vdash$' permits sound steps only, then from "$(\emptyset, \emptyset, G) \vdash^{\oplus} (L, H, \emptyset)$" we may conclude that '$G$' is valid. This idea is formalized in:

Definition 9 Soundness of Inference Step.
The inference step "$(L, H, G) \vdash (L', H', G')$" is called *sound* if validity of 'form$[G']$' implies validity of 'form$[G]$'.

Corollary 10. *If all inference steps in "$(L, H, G) \vdash^{\oplus} (L', H', \emptyset)$" are sound, then 'form$[G]$' is valid.*

Besides the soundness of an inference *step* described above, it is also useful to know about invariant properties of a prover *state* because they can be used to justify why an inference step must be sound. The following such property is most natural, stating that all lemmas are valid and that the hypotheses are founded on the goals, i. e. that for each counterexample for a hypothesis there is a smaller counterexample for a goal.

Definition 11 Correctness of Prover State.
A prover state (L, H, G) is called *correct* if L is valid and $H \curvearrowright G$.

While the first part of this definition should be immediately clear, "$H \curvearrowright G$" states that the goals carry the proof work (which has to be done for the hypotheses) in such a way that the transformation of a goal may make use of hypotheses which are smaller (w. r. t. our induction ordering $<$) than the goal itself since minimal counterexamples for goals cannot be deleted that way. While "correctness of prover states" obviously formulates this idea, it is not the only possible way to do it:

Definition 12 Weak Correctness of Prover State.
A prover state (L, H, G) is called *weakly correct* if
L is valid and $(H \not\swarrow G \;\Rightarrow\; (\mathrm{form}[H] \text{ is valid}))$.

By the corollaries 4 and 2(3) we get:

Corollary 13. *If a prover state is correct, then it is weakly correct, too.*

As announced above, correctness of prover states really permits us to conclude that the inference steps of our frame inference system are sound:

Lemma 14 Soundness of Inference Steps.
If (L, H, G) is a [weakly] correct prover state, then an inference step " $(L, H, G) \vdash (L', H', G')$ " (with the above rules) is sound.

(For a proof cf. Appendix D.) Furthermore, correctness holds indeed for an initial state and is kept invariant by the frame inference system:

As a corollary of corollaries 3 and 13 we get:

Corollary 15 Initial State is Correct.
Let L be valid and $H \subseteq G$. Now (L, H, G) is [weakly] correct.

Lemma 16 Invariance of Correctness of Prover States.
If " $(L, H, G) \vdash (L', H', G')$ " (with the above rules) and the prover state (L, H, G) is [weakly] correct, then '(L', H', G')' is [weakly] correct, too.

(For a proof cf. Appendix D.) Finally, "correctness of prover states" as an invariance property is not only useful to conclude soundness of single steps, but also globally useful, which can be seen in the following corollary stating that the lemmas and hypotheses gathered in a final prover state are valid:

As a corollary of the corollaries 3, 2(1), and 13 we get:

Corollary 17 For Final State: Correctness means Validity.
$(L', H', \emptyset)$ is [weakly] correct iff " $L' \cup \mathrm{form}[H']$ " is valid.

7 The Backwards Approach

The backwards approach was first formalized in Becker (1994).

Definition 18 Inductiveness and Inductive Soundness.
A prover state (L, H, G) is called *inductive* if $((L \text{ is valid}) \;\Rightarrow\; H \not\swarrow G)$.
The inference step " $(L, H, G) \vdash (L', H', G')$ " is called *inductively sound*[3] if inductiveness of (L', H', G') implies inductiveness of (L, H, G).

[3] In Becker (1994) this is called *preservation of* (inductive) *counterexamples*, but we cannot use this name here because the notion of "counterexample" is different there.

Inductiveness is a technical notion abstracted from inference systems similar to the frame inference system of sect. 5. Roughly speaking, inductiveness of a state means that an inductive proof of it is possible in the sense that a final (i. e. successful) prover state can be entailed. This is because the goals can be deleted, since there are either false lemmas (ex falso quodlibet) or false hypotheses below all invalid goals.

Inductive soundness can replace soundness of prover steps, by the following argumentation, which (just like soundness) requires to think ' $\vdash$ ' backwards, starting from a final prover state $(L', H', \emptyset)$, which must be inductive. Now, if the steps deriving a final state are inductively sound, then all states involved must be inductive. Finally, inductiveness of an initial state implies validity of the initial set of goals.

As a corollary of Corollary 3 we get:

Corollary 19 Final State Must Be Inductive. $(L', H', \emptyset)$ *is inductive.*

By Corollary 2(5) for the forward and by the corollaries 3, 4 and 2(3) for the backwards direction we get:

Corollary 20 For Initial State: Inductiveness means Validity of Goals. *Let* L *be valid and* $H \subseteq G$ *. Now,* 'form$[G]$' *is valid iff* (L, H, G) *is inductive.*

By the corollaries 19 and 20 we conclude:

Corollary 21. *If all inference steps in "* $(\emptyset, \emptyset, G) \vdash^{\oplus} (L', H', \emptyset)$ *" are inductively sound, then* 'form$[G]$' *is valid.*

Unlike soundness, inductive soundness also captures the basic idea of our frame inference system which for the analytic approach had to be expressed by some correctness property, namely the idea that transformations of goals may make use of hypotheses which are smaller than the goal itself since minimal counterexamples for goals can never be deleted that way. Note, however, that "being not inductive" is no invariance property of ' $\vdash$ ' (like correctness is) because one never knows whether it holds for some state or not: If all steps are inductively sound, we only know that the property of "being not inductive" is never removed by an inference step, but this does not mean that it ever holds. Especially for successful proofs it never does, cf. Corollary 19. Instead, inductiveness (i. e. "being inductive") is an invariance property of ' $\dashv$ '.

Since inductive soundness captures the basic idea of our frame inference system, we get (cf. Appendix D for a proof):

Lemma 22 Inductive Soundness of Inference Steps.
An inference step with the above rules is inductively sound.

8 Discussion of the Two Approaches

While the relation between the two approaches of the previous two sections is not simple, both seem to be equally useful in capturing the ideas presented in the beginning of sect. 2 as well as in explaining the soundness of our inference system: The following is a corollary of 10, 14, 15, & 16, as well as a corollary of 21 & 22:

Corollary 23 Soundness of " $(\emptyset,\emptyset,G) \vdash^{\oplus} (L',H',\emptyset)$ ".
If " $(\emptyset,\emptyset,G) \vdash^{\oplus} (L',H',\emptyset)$ " (with the above rules), then " $\mathrm{form}[G]$ " is valid.

The analytic approach even permits a slightly stronger conclusion via Corollary 17:

Corollary 24. *If " $(\emptyset,\emptyset,G) \vdash^{\oplus} (L',H',\emptyset)$ " (with the above rules), then " $\mathrm{form}[G] \cup L' \cup \mathrm{form}[H']$ " is valid.*

Considering the design of concrete inference systems by presenting sub-rules of the frame inference rules, another advantage of the analytic approach could be that the additional assumption of correctness of the states could be essential for the sub-rule relationship.

Finally, we compare the analytic and the backwards approach independently of our frame inference system. Here, we consider one approach to be superior to the other, when it permits additional successful proofs, whereas we do not respect the fact that one notion may be more appropriate for effective concretion than the other. Invariance of correctness cannot be superior to invariance of weak correctness or to inductive soundness, since a step with the former and without one of the latter properties starts from an invalid set of goals and thus the required soundness of inference steps does not permit additional proofs. Invariance of weak correctness cannot be superior to invariance of correctness (or else to inductive soundness), since a step with the former and without one of the latter properties leads to (or else starts from) an invalid set of goals and thus the required soundness of inference steps does not permit additional proofs. Finally, inductive soundness is very unlikely to be superior to invariance of [weak] correctness, since a step with the former and without one of the latter properties leads to an invalid set of lemmas or hypotheses. Moreover, if we do not consider all proofs but only the existence of proofs, then (on our non-effective level!) all approaches are equivalent: Using the Deletion rule $|G|$ -times we get:

Corollary 25 Completeness of " $(\emptyset,\emptyset,G) \vdash^{\oplus} (L',H',\emptyset)$ ".
If " $\mathrm{form}[G]$ " is valid, then " $(\emptyset,\emptyset,G) \vdash^{\oplus} (\emptyset,\emptyset,\emptyset)$ " (with the above rules).

Note, however, that (as far as we know) for the construction of effective concrete inference systems based on rules which are no (effective) sub-rules of the rules of our frame inference system, each of the three approaches (i. e.: soundness and invariance of correctness; soundness and invariance of weak correctness; inductive soundness) may be superior to each of the others. The same may hold for generalizing them to inference systems on generalized prover states. E. g., for

Parallelization: $\dfrac{(L,H,G \cup G')}{(L,H,G) \qquad (L,H,G')}$

it is obvious how to generalize inductive soundness, whereas the two other approaches do not seem to permit an appropriate generalization based on local properties of triples (L,H,G).

9 The "Switched" Frame Inference System

In sect. 2 we pointed out that we have to avoid non-terminating reasoning cycles between hypotheses and goals. In our formalization we achieved this by founding a hypothesis '$\aleph$' on smaller or equal goals from 'G', i. e. " $\{\aleph\}\curvearrowright G$ " (cf. the condition of the Hypothesizing rule), and by applying to a goal '$\gimel$' only strictly smaller hypotheses from 'H', i. e. " $\{\gimel\}\searrow H$ " (cf. the condition of the Deletion rule). From a cyclic reasoning " $H\curvearrowright G\searrow H$ " we immediately get " $H\searrow H$ " and " $G\searrow G$ " by Corollary 4, and then 'form[$H\cup G$]' is valid by Corollary 2(3), which means that the reasoning cycle is sound. Now an alternative way to achieve this is the following: Instead of doing our quasi-decreasing step '$\curvearrowright$' from hypotheses to goals " $H\curvearrowright G$ " and our strictly decreasing step '$\searrow$' from goals to hypotheses " $G\searrow H$ ", we could go from hypotheses to goals with a strict and from goals to hypotheses with a quasi step. More precisely: The condition of the Hypothesizing rule would be changed into " if L is invalid or $\{\aleph\}\ \searrow/\curvearrowright\ (G, H)$ ", and the condition of the Deletion rule is changed into " if L is invalid or $\{\gimel\}\curvearrowright G\cup H$ ". The Expansion and the Acquisition rules remain unchanged. A Memorizing sub-rule cannot exist and the Transformation rule must be composed of several Expansions, an optional following Hypothesizing, and then a Deletion into one of the following forms:

Memorizing Switched Transformation:
$$\frac{(L\ ,H\qquad\qquad ,G\cup\{\,\aleph\,\}\,)}{(L\ ,H\cup\{\,\aleph\,\}\,,G\cup M\quad\ )}$$
if L is invalid or $\{\aleph\}\ \searrow/\curvearrowright\ (G\cup M, H)$

Simple Switched Transformation:
$$\frac{(L\ ,H\qquad ,G\cup\{\,\aleph\,\}\,)}{(L\ ,H\qquad ,G\cup M\quad\ )}$$
if L is invalid or $\{\aleph\}\curvearrowright G\cup M\cup H$

When we then also switch '$\curvearrowright$' and '$\searrow$' (i. e. replace one by the other) in the definitions of "correctness" and "inductiveness" and when we require an initial state additionally to have an empty set of hypotheses, then we get the analogous results for the soundness of our *switched* frame inference system. The reasons why we prefer the non-switched version presented here are the following:

From a user's point of view, the non-switched version may be more convenient, because hypotheses become available earlier and easier via the Memorizing rule, which does not exist for the switched version.

From the inference system designer's point of view, the non-switched version is more convenient, due to the following argumentation: With both the switched and the non-switched version of the inference system, a proof can be thought to consist mainly of steps of the kind that a goal '$\aleph$' may become available as a hypothesis and is then transformed into sub-goals 'M'. For the non-switched case this can be achieved by an application of the Memorizing and then of the Transformation rule. For the switched inference system this is just a Memorizing Switched Transformation. One shortcoming of the switched version results from the fact that the transformation of a goal into sub-goals has to be strictly decreasing instead of quasi-decreasing (as required for the non-switched case). The design of quasi-decreasing transformations, however, is easier than that of strictly decreasing ones, for the same reason as exhibited in sect. 2 ("Easier Design of Inference Rules") and as illustrated in Appendix C. Therefore, the non-switched inference system allows for small grain inference steps which in the switched system must be replaced with a very big inference step bridging over all quasi-decreasing steps until a strictly decreasing step is reached. Another shortcoming of the Memorizing Switched Transformation is that each simplification step has to decrease the weight of the goal strictly, i. e. that the possibility to apply some fixed smaller hypothesis gets more and more unlikely with each simplification step, cf. sect. 2 ("High Quality of Ordering Information"). If we, however, use the Simple Switched Transformation instead, then the goal is not made available as a hypothesis. Thus, in order not to lose the possibility to apply hypotheses, simplification should not be done via inference steps of the switched inference system, but incorporated into the hypotheses applicability test. With the non-switched version, however, the full inference power of the whole inference system can be homogeneously used for simplification.

On the other hand, the comparison of weights for an applicability test of a hypothesis to a goal is simpler in the switched inference system because there the weight of the hypothesis is often equal to the weight of the goal, in which case the test is successful. While this does not allow for additional proofs with the switched inference system, it may allow to avoid possibly complicated reasoning on ordering properties.

10 Classifying Other Work

In this section we give an incomplete sketch of the literature on inference systems for implicit induction and briefly classify these inference systems according to our presentation here.

In Bachmair (1988) our sets of hypotheses and goals are not separated yet. A disadvantage of this is that a success of a proof due to an empty set of goals is more difficult to detect and that understanding the inference system gets more difficult without the concepts of hypotheses and goals. The missing separation into hypotheses and goals also requires both the foundedness step from hypotheses to goals (as in our switched system of sect. 9) and the step from goals to hypotheses (as in the non-switched system of sect. 5) to be strictly decreasing (i. e. '$\searrow$'), which means a combination of the disadvantages of both the switched and the non-switched approach. The soundness of a proof in Bachmair's inference system results from the fact that a fair derivation sequence $M_i \vdash M_{i+1}$ ($i \in \mathbb{N}$) always satisfies $M_i \curvearrowright M_{i+1}$ and has a sub-sequence such that $\forall i \in \mathbb{N}$: $M_{j_i} \searrow M_{j_{i+1}}$, which means that 'form$[M_0]$' must be valid, cf. corollaries 8, 7, and 2(6). The foundedness relations are defined by use of sizes of inconsistency proofs for the equations in M_i instead of the counterexamples themselves. As already mentioned in sect. 3, it is in general undesirable to base the notions of a frame inference system on formal proofs instead of semantic notions because the latter impose no operational restrictions and are likely to change less frequently. One of the operational restrictions in Bachmair (1988) is the confluence requirement for the specifying set of rules. That this restriction can be removed was noted in Gramlich (1989) by defining the foundedness relations by use of the sizes of positive proofs measured via the applications of equations from M_i.

The important separation between hypotheses and goals was introduced in Reddy (1990), where a frame inference system similar to our switched one in sect. 9 is used, the argumentation for soundness follows the analytic approach using operationally restricted versions of soundness and (switched) correctness, and the foundedness notions are still the operationally restricted ones of Gramlich (1989). In Wirth (1991) this operational restriction is overcome by using a semantic foundedness notion.

In Fraus (1993) we have found the first[4] argumentation for soundness following the backwards approach. While the inference system already is of the (superior) non-switched style, the foundedness relations are still operationally restricted (by measuring positive proofs in the natural deduction calculus). In Becker (1994) we finally find the backwards approach based on semantic foundedness notions as presented here.

11 Conclusion

We tried to give an intuitive understanding of proofs by mathematical induction, exhibited the essential requirements, and provided a simple data structure for prover states. To enable a clear understanding of the functions of inference systems for proofs by mathematical induction, we introduced the concept of "foundedness" which also has applications beyond this paper. We presented an abstract frame inference system, elaborated two approaches for explaining why this system is sound, and argued why we prefer our frame inference system to the switched one. We classified argumentation for soundness occurring in the literature according to our taxonomy. When practically appropriate concrete inference systems are designed as systems of sub-rules of the rules of the presented frame inference system, soundness of these systems is given immediately. While we did not present a concrete inference system in this paper, the example in Appendix C should be sufficient to make our intention obvious.

Appendix A Safe Steps and Failure Recognition

As already mentioned in sect. 1, the ability of an inductive theorem prover to detect invalid formulas is most important under a practical aspect, especially for *inductive* theorem proving because of the important role generalizations play in it, where an invalid formula can result from a valid input theorem due to overgeneralization. Thus, suppose we have some failure predicate 'FAIL' defined on sets of formulas which is *correct* in the sense that $\forall F \in \mathrm{FAIL}$: ($F$ is invalid). Note that this failure predicate is defined on *sets of* formulas for operational

[4] This actually goes back to an unpublished manuscript of Alfons Geser (1988) at the University of Passau entitled "An inductive proof method based on narrowing".

reasons, namely in order to be able to recognize that one formula contradicts another one; whereas for a theoretical treatment it would be sufficient to define it on single formulas since one of those formulas must be invalid in a consistent specification; but to find out which formula it is is undecidable in general.

We define a prover state (L, H, G) to be a *failure state* if

$$(L \cup \mathrm{form}[H \cup G]) \in \mathrm{FAIL}.$$

Note that we have included L and H (instead of just testing G) because we want to be able to detect an invalid lemma or hypothesis when it has just been generated and do not want to have to wait until it will have been harmfully applied to a goal. One is tempted to argue that, in case of [weak] correctness of a prover state, an invalid hypothesis implies the existence of an invalid goal, but this argument again does not respect the operational aspect.

Now, when an inductive theorem prover has realized to be in a failure state, the following questions arise: How far do we have to backtrack to reach a state with valid formulas? Have some of our original input formulas been invalid? For answering these questions the following notion is useful:

An inference step " $(L, H, G) \vdash (L', H', G')$ " is called *safe* if

$$\text{validity of " } L \cup \mathrm{form}[H \cup G] \text{ " implies validity of " } L' \cup \mathrm{form}[H' \cup G'] \text{ ".}$$

It is not reasonable to require all possible steps of an inductive theorem prover to be safe, since this property is undecidable for generalization steps which play a major role in inductive theorem proving. For concrete inference systems, however, it is usually possible to give interesting sufficient conditions for the application of an inference rule to be safe. Now, when the prover has found out that a prover state (L'', H'', G'') is a failure state and all steps in $(L', H', G') \vdash^{\circledast} (L'', H'', G'')$ are known to be safe, then (L', H', G') must be a failure state, too. To recover from this failure we may iterate the following:

If this (L', H', G') is the original input state (with L' known to be valid and $H' \subseteq G'$), then we have refuted our original set of goals G' and should stop proving. Otherwise the step that yielded (L', H', G'), say $(L, H, G) \vdash (L', H', G')$, must be carefully inspected: If it is known to be safe we backtrack this step and reiterate. Otherwise it might be possible to find a (minimal) subset of $L'' \cup \mathrm{form}[H'' \cup G'']$ for which the failure predicate FAIL is still known to hold and which also is (implied by) a subset of $L \cup \mathrm{form}[H \cup G]$; in which case we also backtrack this step and reiterate. Otherwise, when $(L, H, G) \vdash (L', H', G')$ is likely to be an unsafe step which might have caused the failure, we backtrack this step and may try to go on with a hopefully safe inference step instead.

Appendix B Refutational Completeness

For achieving refutational completeness we need a wellfounded ordering $>_{\mathrm{refut}}$ on finite sets of syntactic constructs. To be able to refute initial failure states we need the following property.

Definition 26 **FAIL-Completeness.**
The failure predicate FAIL is *complete w. r. t. '* $\vdash$ *' and '*$>_{\mathrm{refut}}$*'* if for all finite sets $L \subseteq \mathrm{Form}$; $H, G \subseteq \mathrm{SynCons}$; if $\mathrm{form}[G]$ is invalid, but (L, H, G) is not a

failure state, then there are finite sets L', H', G' with $(L,H,G) \vdash^{\oplus} (L',H',G')$ and $G >_{\mathrm{refut}} G'$.

By wellfoundedness of '$>_{\mathrm{refut}}$' we immediately get:

Corollary 27 Refutational Completeness.
Let $L \subseteq$ Form; $H, G \subseteq$ SynCons be finite sets. Assume either that '$\vdash$' is sound or that L is valid, $H \subseteq G$, and '$\vdash$' is inductively sound. Furthermore assume FAIL *to be complete w. r. t. '$\vdash$' and '$>_{\mathrm{refut}}$'. Now, if* form$[G]$ *is invalid, then there is some failure state (L',H',G') with $(L,H,G) \vdash^{\oplus} (L',H',G')$.*

Definition 28 Fairness.
Let β be an ordinal number with $\beta \preceq \omega$. Let $L_i \subseteq$ Form; $H_i, G_i \subseteq$ SynCons for all $i \prec 1{+}\beta$. Consider the derivation $(L_i,H_i,G_i) \vdash (L_{i+1},H_{i+1},G_{i+1})$ $(i \prec \beta)$. It is called *fair* if $\beta \prec \omega \wedge (L_\beta,H_\beta,G_\beta) \notin \mathrm{dom}(\vdash)$ (i. e. no inference rule can be applied to $(L_\beta,H_\beta,G_\beta)$) or $\exists i{\prec}1{+}\beta\colon\ G_i = \emptyset$ or $\forall i{\prec}1{+}\beta\colon$ $\big((\mathrm{form}[G_i]$ invalid $\wedge$ (L_i,H_i,G_i) not a failure state$)$ $\Rightarrow$ $\exists j{\prec}1{+}\beta\colon\ G_i >_{\mathrm{refut}} G_j\big)$.

Corollary 29. *Let β be an ordinal number with $\beta \preceq \omega$. Let $L_i \subseteq$ Form; $H_i, G_i \subseteq$ SynCons be finite sets for all $i \prec 1{+}\beta$. Let $(L_i,H_i,G_i) \vdash (L_{i+1},H_{i+1},G_{i+1})$ $(i \prec \beta)$ be a fair derivation. Assume either that '$\vdash$' is sound or that L_0 is valid, $H_0 \subseteq G_0$, and '$\vdash$' is inductively sound. Furthermore assume* FAIL *to be complete w. r. t. '$\vdash$' and '$>_{\mathrm{refut}}$'. Now, if* form$[G]$ *is invalid, there must exist some $i \prec 1{+}\beta$ such that (L_i,H_i,G_i) is a failure state.*

Appendix C An Example

In this section we give an example to illustrate our abstract argumentation on the benefit of separating weights from formulas and of using non-switched inference systems. Consider the following specification of a member-predicate "$\mathsf{mbp}(x,l)$" testing for x occurring in the list l, a delete-function "$\mathsf{dl}(x,l)$" deleting all occurrences of x in the list l, a remove-copies-function "$\mathsf{rc}(x,l)$" removing repeated occurrences of x in the list l, and a brushing function "$\mathsf{br}(k,l)$" removing repeated occurrences in the list l for all elements of the list k:

$\mathsf{mbp}(x,\mathsf{nil})$	$=\mathsf{false}$		$\mathsf{dl}(x,\mathsf{cons}(y,l))$	$=\mathsf{cons}(y,\mathsf{dl}(x,l))$	**if** $x{\neq}y$
$\mathsf{mbp}(x,\mathsf{cons}(y,l))$	$=\mathsf{true}$	**if** $x{=}y$	$\mathsf{rc}(x,\mathsf{nil})$	$=\mathsf{nil}$	
$\mathsf{mbp}(x,\mathsf{cons}(y,l))$	$=\mathsf{mbp}(x,l)$	**if** $x{\neq}y$	$\mathsf{rc}(x,\mathsf{cons}(y,l))$	$=\mathsf{cons}(y,\mathsf{dl}(x,l))$	**if** $x{=}y$
$\mathsf{dl}(x,\mathsf{nil})$	$=\mathsf{nil}$		$\mathsf{rc}(x,\mathsf{cons}(y,l))$	$=\mathsf{cons}(y,\mathsf{rc}(x,l))$	**if** $x{\neq}y$
$\mathsf{dl}(x,\mathsf{cons}(y,l))$	$=\mathsf{dl}(x,l)$	**if** $x{=}y$	$\mathsf{br}(\mathsf{nil},l)$	$=l$	
			$\mathsf{br}(\mathsf{cons}(x,k),l)$	$=\mathsf{br}(k,\mathsf{rc}(x,l))$	

Suppose we want to show the following theorem (", " denotes "logical or"):

(0) $\mathsf{br}(k,\mathsf{cons}(x,l)) = \mathsf{cons}(x,\mathsf{br}(k,l)),\ \ \mathsf{mbp}(x,l) = \mathsf{true}$

saying that either x occurs in l or the wave-front $\mathsf{cons}(x,\ldots)$ can be rippled out. Applying a covering set of substitutions to (0) we get a base case for $\{\, k \mapsto \mathsf{nil} \,\}$ (which is trivial after two rewriting steps applying the first rule for br) and the following case for $\{\, k \mapsto \mathsf{cons}(y,k) \,\}$:

(1) $\mathsf{br}(\mathsf{cons}(y,k),\mathsf{cons}(x,l)) = \mathsf{cons}(x,\mathsf{br}(\mathsf{cons}(y,k),l)),\ \ \mathsf{mbp}(x,l) = \mathsf{true}$

After two rewriting steps applying the second rule for br we get:

(2) $\mathsf{br}(k, \mathsf{rc}(y, \mathsf{cons}(x, l))) = \mathsf{cons}(x, \mathsf{br}(k, \mathsf{rc}(y, l))),\ \ \mathsf{mbp}(x, l) = \mathsf{true}$

A rewriting step applying the third rule for rc in the left-hand side of the first equation yields

(3) $x = y,\ \ \mathsf{br}(k, \mathsf{cons}(x, \mathsf{rc}(y, l))) = \mathsf{cons}(x, \mathsf{br}(k, \mathsf{rc}(y, l))),\ \ \mathsf{mbp}(x, l) = \mathsf{true}$

as well as the goal

(G) $x \neq y,\ \ \mathsf{br}(k, \mathsf{rc}(y, \mathsf{cons}(x, l))) = \mathsf{cons}(x, \mathsf{br}(k, \mathsf{rc}(y, l))),\ \ \mathsf{mbp}(x, l) = \mathsf{true}$

whose proof we do not treat here.

Applying our induction hypothesis (0) instantiated by $\{\, l \mapsto \mathsf{rc}(y, l)\,\}$, i. e.

(I) $\mathsf{br}(k, \mathsf{cons}(x, \mathsf{rc}(y, l))) = \mathsf{cons}(x, \mathsf{br}(k, \mathsf{rc}(y, l))),\ \ \mathsf{mbp}(x, \mathsf{rc}(y, l)) = \mathsf{true}$

to (3) we get:

(4) $\mathsf{mbp}(x, \mathsf{rc}(y, l)) \neq \mathsf{true},$

$x = y,\ \ \mathsf{br}(k, \mathsf{cons}(x, \mathsf{rc}(y, l))) = \mathsf{cons}(x, \mathsf{br}(k, \mathsf{rc}(y, l))),\ \ \mathsf{mbp}(x, l) = \mathsf{true}$

Rewriting with the lemma $\mathsf{mbp}(x, \mathsf{rc}(y, l)) = \mathsf{mbp}(x, l)$ finally yields a tautology.

For this induction proof to be sound we have to find a wellfounded ordering in which the instance (I) of our hypotheses is (strictly) smaller than the goal (3) to which it is applied. When we consider the weight of a formula to be the formula itself (considered as the multi-set of (its literals considered as the multi-sets of) its terms), this cannot be achieved with a simplification ordering, whereas the evaluation ordering of the specification is a sub-relation of a simplification ordering, namely the lexicographic path ordering given by the following precedence on function symbols: $\mathsf{mbp} \gtrsim \mathsf{true}, \mathsf{false};\ \ \mathsf{br} \gtrsim \mathsf{rc} \gtrsim \mathsf{dl} \succ \mathsf{cons}$. The first thing we can do is to use weight pointers to avoid the deterioration of ordering information on the way from (1) to (3): Initially the weight pointer of (0) points to (0) itself, i. e. the weight of this formula is this formula itself. The same holds for (1) because when applying the substitution the weight is instantiated as well as the formula. During the simplification steps yielding (2) and then (3) the weight remains unchanged, i. e. the weight of the formula (3) is still the formula (1). Now (I) is indeed strictly smaller than (1) in the lexicographic path ordering given by: $\mathsf{br} \succ \mathsf{cons};\ \ \mathsf{br} \gtrsim \mathsf{rc};\ \ \mathsf{br} \succ \mathsf{mbp}$. Thus the high quality of ordering information preserved by separating the formula from its weight when simplifying (1) into (3) now permits us to justify the application of (I) to (3) with a simplification ordering, i. e. we can now realize that our (partial) proof is sound. This success, however, is not very convincing because the last pair in the above precedence (i. e. $\mathsf{br} \succ \mathsf{mbp}$) is not all motivated by the evaluation ordering of our specification. After all, our example proof is nothing but a structural induction and thus should work out without such a sophisticated ordering. If we have a closer look on the derivation from (0) (or (1)) to (3), then we notice that the first literal as usual does get smaller in the evaluation ordering on the way from our original goal (over the goal the hypothesis is applied to) to the instantiated hypothesis (I), but the second literal does not. Thus, if we focus on the first literal by setting our weight to it, then we only have to show that the first literal of (I) is smaller than the first literal of (1) which does not require the unmotivated precedence of $\mathsf{br} \succ \mathsf{mbp}$. Going one step further, setting the weight of (0) to be the variable k, the weight of (1) and (3) becomes $\mathsf{cons}(y, k)$ which is trivially greater than the weight k of (I).

The proof of (G) gets easier when we have the following lemma:
(L) $\quad \mathsf{dl}(x,l) = l, \;\; \mathsf{mbp}(x,l) = \mathsf{true}$
We will now use this lemma to show why the design of inference rules gets easier when we separate the weight of a formula from the formula itself and when we do not use the switched inference system of sect. 9 instead of the non-switched one of sect. 5. Suppose we want to apply (L) in the proof of a formula
(10) $\quad \Gamma[\mathsf{dl}(x,l)]$ $\qquad$ containing $\mathsf{dl}(x,l)$ as a subterm.
Rewriting with (L) yields the following two sub-goals:
(11) $\quad \mathsf{mbp}(x,l) = \mathsf{true}, \;\; \Gamma[l]$ $\qquad$ (12) $\quad \mathsf{mbp}(x,l) \neq \mathsf{true}, \;\; \Gamma[\mathsf{dl}(x,l)]$
When we restrict the weight of a formula to be the formula itself, then the subgoals (11) and (12) must be smaller than (10) (without focusing on $\Gamma[\mathsf{dl}(x,l)]$). When we choose the switched inference system and want to make (10) available as an induction hypotheses, then the weights of the sub-goals (11) and (12) must be strictly smaller than the weight (10) in a wellfounded ordering. For (11) this might be achieved again by the unmotivated trick of extending the precedence on function symbols with $\mathsf{dl} \succ \mathsf{mbp}$ (additionally to $\mathsf{mbp} \succsim \mathsf{true}$); for (12), however, this does not seem to be reasonably possible in general. Thus the design of an inference rule applying (L) in the intended form to (10) must be very difficult to develop without separated weights or for a switched frame inference system, because the step from (10) to (12) must be replaced with a very big inference step bridging over all steps following in the proof of (12) until all branches of this proof have reached a smaller weight. Separating weights from formulas and using a non-switched frame inference system, however, the design of such an inference rule is very easy: One just sets the weight of (11) and (12) to the the original weight of formula (10).

Appendix D The Proofs

Proof of Lemma 14: The only rule whose soundness is non-trivial is the Deletion rule for form$[G]$ being valid. By Corollary 2(2) we get $G \searrow \emptyset$. Thus by Corollary 6(2) and the condition of the rule we get $\{\gimel\} \searrow H$. By Corollary 13 $(L, H, G \cup \{\gimel\})$ is weakly correct. Thus (since $G \cup \{\gimel\} \searrow H$ due to Corollary 5(2)) we can conclude that form$[H]$ must be valid. By Corollary 2(7) this means that form$(\gimel)$ is valid, i. e. that form$[G \cup \{\gimel\}]$ is valid.

Proof of Lemma 16: Assume L to be valid. We show invariance of weak correctness first: <u>Expansion:</u> Assume $G \cup \{\gimel\} \searrow H$. By Corollary 5(4) we get $G \searrow H$. Now form$[H]$ must be valid due to weak correctness of (L, H, G). <u>Hypothesizing:</u> Assume $G \searrow H \cup \{\aleph\}$. By Lemma 22 we get $G \searrow H$. By weak correctness of (L, H, G), form$[H]$ must be valid, which by Corollary 2(7) means that form$[G \cup H]$ is valid. By the condition of the rule and Corollary 2(6), we know that form$(\aleph)$ is valid. Therefore form$[H \cup \{\aleph\}]$ is valid, too. <u>Acquisition:</u> Trivial. <u>Deletion:</u> Assume $G \searrow H$. By Lemma 22 we get $G \cup \{\gimel\} \searrow H$. By weak correctness of $(L, H, G \cup \{\gimel\})$, form$[H]$ must be valid.

Finally we show invariance of correctness: <u>Expansion:</u> By Corollary 5(3). <u>Hypothesizing:</u> By Corollary 7 we get $G \curvearrowright G$. If we assume $H \curvearrowright G$, we get

$H{\cup}G{\curvearrowright}G$ by Corollary 5(1). By the condition of the rule and Corollary 7 this means $\{\aleph\}{\curvearrowright}G$. By our assumption we now get $H{\cup}\{\aleph\}{\curvearrowright}G$ via Corollary 5(1). Acquisition: Trivial. Deletion: Assume $H{\curvearrowright}G{\cup}\{\gimel\}$. By Corollary 6(1) from the condition of the rule we get $\{\gimel\}{\curvearrowright}G$. By Corollary 7 we have $G{\curvearrowright}G$ and then by Corollary 5(1) $G{\cup}\{\gimel\}{\curvearrowright}G$. By the assumption and Corollary 7 this means $H{\curvearrowright}G$.

Proof of Lemma 22: Expansion: By Corollary 5(4). Hypothesizing: By Corollary 7 we get $H{\curvearrowright}H$ and then from the condition of the rule $H{\cup}\{\aleph\} \curvearrowright H{\cup}G$ by Corollary 5(1). If we now assume $G{\searrow}H{\cup}\{\aleph\}$, we get $G{\searrow}H{\cup}G$ by Corollary 3, and then by corollaries 5(5) and 5(4) $G{\searrow}H$. Acquisition: Trivial. Deletion: Assume $G{\searrow}H$. By the condition of the rule and Corollary 6(2) we get $\{\gimel\}{\searrow}H$. Thus by our assumption and Corollary 5(2) we get $G{\cup}\{\gimel\}{\searrow}H$.

Acknowledgements: We would like to thank Jürgen Avenhaus, Alfons Geser, Bernhard Gramlich, Ulrich Kühler, and Martin Protzen for useful hints.

References

Leo Bachmair (1988). *Proof by Consistency in Equational Theories.* 3rd IEEE symposium on Logic In Computer Science, pp. 228-233.

Klaus Becker (1993). *Proving Ground Confluence and Inductive Validity in Constructor Based Equational Specifications.* TAPSOFT 1993, LNCS 668, pp. 46-60, Springer.

Klaus Becker (1994). *Rewrite Operationalization of Clausal Specifications with Predefined Structures.* PhD thesis, Fachbereich Informatik, Universität Kaiserslautern.

Wolfgang Bibel, E. Eder (1993). *Methods and Calculi for Deduction.* In: Dov M. Gabbay, C. J. Hogger, J. A. Robinson (eds.). *Handbook of Logic in Artificial Intelligence and Logic Programming.* Vol. 1, pp. 67-182, Clarendon.

Robert S. Boyer, J Strother Moore (1979). *A Computational Logic.* Academic Press.

Ulrich Fraus (1993). *A Calculus for Conditional Inductive Theorem Proving.* 3rd CTRS 1992, LNCS 656, pp. 357-362, Springer.

Bernhard Gramlich (1989). *Inductive Theorem Proving Using Refined Unfailing Completion Techniques.* SEKI-Report SR–89–14 (SFB), Fachbereich Informatik, Universität Kaiserslautern. *Short version in:* 9th ECAI 1990, pp. 314-319, Pitman.

Martin Protzen (1994). *Lazy Generation of Induction Hypotheses.* 12th CADE 1994, LNAI 814, pp. 42-56, Springer.

Uday S. Reddy (1990). *Term Rewriting Induction.* 10th CADE 1990, LNAI 449, pp. 162-177, Springer.

Christoph Walther (1994). *Mathematical Induction.* In: *Handbook of Logic in Artificial Intelligence and Logic Programming.* eds. cf. above. Vol. 2, pp. 127-228, Clarendon.

Claus-Peter Wirth (1991). *Inductive Theorem Proving in Theories specified by Positive/Negative Conditional Equations.* Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern.

Claus-Peter Wirth, Bernhard Gramlich (1993). *A Constructor-Based Approach for Positive/Negative-Conditional Equational Specifications.* 3rd CTRS 1992, LNCS 656, pp. 198-212, Springer. Revised and extended version in J. Symbolic Computation (1994) **17**, pp. 51-90, Academic Press.

Claus-Peter Wirth, Bernhard Gramlich (1994). *On Notions of Inductive Validity for First-Order Equational Clauses.* 12th CADE 1994, LNAI 814, pp. 162-176, Springer.