On Notions of Inductive Validity for First-Order Equational Clauses*

Claus-Peter Wirth and Bernhard Gramlich

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

Abstract. We define and discuss various conceivable notions of inductive validity for first-order equational clauses. This is done within the framework of constructor-based positive/negative conditional equational specifications which permits to treat negation and incomplete function definitions in an adequate and natural fashion. Moreover, we show that under some reasonable assumptions all these notions are monotonic w. r. t. consistent extension, in contrast to the case of inductive validity for initial semantics (of unconditional or positive conditional equations). In particular from a practical point of view, this monotonicity property is crucial since it allows for an incremental construction process of complex specifications where consistent extensions of specifications cannot destroy the validity of (already proved) inductive properties. Finally we show how various notions of inductive validity in the literature fit in or are related to our classification.

1 Introduction, Motivation, and Overview

Given some finite axiomatization R, e. g., by means of a set of equations or — more generally — of first-order formulas, one is often not only interested in those properties that are logical consequences of R, i. e. that hold in all models of R but also in properties that only hold in some specific intended model (or class of models) specified by R. Instead of restricting the class of models by some required property, one may also define notions of validity by saying that a formula $P(\mathbf{x})$ is to hold schematically in the sense that it is to hold for certain sets of instantiations for the variables $\mathbf{x}$ of $P(\mathbf{x})$.

Concerning the adequacy of a notion of inductive validity we think that there are at least three important criteria.

Coincidence with intuition: It should capture the intention of the human specifier as close as possible.

Monotonicity behaviour: Whenever we extend a specification in some consistent manner then previously valid formulas should still be valid w. r. t. the extended specification.

* This research was supported by "Deutsche Forschungsgemeinschaft, SFB 314 (D4)".

Operational feasibility: It should be operationally feasible in the sense that there are operational characterizations or at least sufficient operational criteria which provide the basis for corresponding theorem proving techniques.

For specifications with unconditional (or positive conditional) equations it is well-known how to obtain initial semantics and how to prove inductive theorems, i. e. equations that hold in the initial model (cf. e. g. [Bac88]). Consider for instance the following specification R over the natural numbers where addition $(+)$ is defined in terms of zero (0) and successor (s).

$$R: \quad 0 + y = y\,,\ s(x) + y = s(x + y)\,.$$

Here it is easy to show by standard techniques (cf. e. g. [BM79], [Bac88]) that

$$x + 0 = x$$

is an inductive theorem, i. e. holds in the initial model. If we now enrich the above specification by a subtraction operation $(-)$ with

$$x - 0 = x\,,\ s(x) - s(y) = x - y\,,$$

yielding a new specification R', then, unfortunately, $x + 0 = x$ is no longer inductively valid in the enriched specification (to wit, substitute $0 - s(0)$ for x). Hence, initial semantics does not enjoy the above-mentioned monotonicity property w. r. t. consistent extension. The intuitive reason for that phenomenon is that by enriching our basic specification R as described above we have introduced new *junk terms* which should not be considered for verification purposes. This intention will be formally reflected in our approach by distinguishing between *constructor* and *general* variables which permits to refine the class of (ground) instances of a theorem that we would like to hold.

Another crucial problem with initial semantics has to do with the way that negative statements are interpreted. For instance, the negative equation

$$0 - s(0) \neq 0$$

holds in the initial model of R'. But if we now complete the partial definition of subtraction in some *consistent* manner, e. g., by adding

$$0 - s(y) = 0$$

yielding the new specification R'', then $0 - s(0) \neq 0$ does not hold any more in the initial model of R''. Of course, also validity of conditional statements or general first-order clauses is influenced by this phenomenon. One may argue now that in R' we are not yet sure whether $0 - s(0) \neq 0$ is to hold because it depends on the way we might (consistently) complete the definition of '$-$' later on in the specification process. Thus, the existence of negative literals in formulas to be proved opens up — or even necessarily entails — various ways of defining inductive validity, in particular if we want to guarantee some reasonable monotonicity property w. r. t. consistent extension. Pioneering papers along this line of reasoning are [KM87] and [KM86] (cf. also [Zha88], [ZKK88]). But whereas in these papers the specifications treated are systems of unconditional equations, and the formulas considered are pure equations, here we shall permit general equational first-order clauses as formulas and, moreover, as specifications we admit positive/negative conditional equational systems which naturally arise in many cases.

Before going into details, let us give a rough idea of our underlying specification formalism using constructor-based positive/negative conditional equations. The basic characteristics of our approach (as developed in [WG93]) may be summarized as follows: The set of function symbols is partitioned into *constructor* and *non-constructor* function symbols. The non-constructor function symbols can be used for (possibly partially) defining functions on a domain of discourse supplied by the constructor (ground) terms and called the *constructor sub-universe*. For such partial definitions of functions, variables ranging only over the constructor terms are useful because sometimes the specifier does not want to prescribe how the functions have to behave on objects that are *undefined* in the sense that they do not belong to the domain of discourse. Therefore, in addition to the usual *general variables* we also permit *constructor variables*. As axiomatizations we consider sets of positive/negative conditional equations of the form $\bigwedge_{i=1}^{m}(s_i = t_i) \wedge \bigwedge_{j=1}^{n}(u_j \neq v_j) \quad \Rightarrow \quad l = r$.
In general, specifications with such positive/negative conditional equations lack a unique minimal (i. e. an initial) model. In [Kap87] an operational semantics is developed which — under some reasonable assumptions — distinguishes one of the minimal models by extracting control information from the equations (considered as rewrite rules). In contrast to [Kap87] we use two syntactically expressible restrictions on the form of the equations/rules in order to obtain an appropriate semantics: Namely, the terms in the negative conditions must be *defined* and the *constructor rules* are required to have *Horn*-form and to be *constructor-preserving* (see below). For such positive/negative conditional rule systems a reduction relation can be defined which is monotonic w. r. t. consistent extension and which provides an operational characterization of a unique minimal model.

As formulas to be proved we consider (implicitly universally quantified) first-order clauses of the form $A_1 \wedge \ldots \wedge A_m \quad \longrightarrow \quad B_1 \vee \ldots \vee B_n$ where the A_i's and the B_j's are atoms over the predicate symbols '$=$' (equality) and 'Def' (definedness predicate) on terms (with general and constructor variables). Some interesting types of inductive validity of such formulas w. r. t. a positive/negative conditional rule system R are defined by restricting the class of models to be considered. Other interesting types of inductive validity are defined by means of *inductive substitutions*, i. e., substitutions that substitute constructor ground terms for constructor variables and leave general variables invariant. The latter invariance for general variables corresponds to the intuition that general variables should be permitted to range not only over the *junk* generated by general ground terms but also over additional *junk*, e. g., of non-constructor symbols which might be introduced later on. Indeed, permitting this additional *junk* is necessary for the intended monotonicity of the notions of inductive validity w. r. t. consistent extension of the specification.

A more detailed discussion of the subject (as well as missing proofs which we have to omit here due to lack of space) can be found in [WGKP93].

2 Preliminaries

2.1 Basic Notions and Notations

We assume familiarity with the basic notions and notations for syntax and semantics of (conditional) term rewriting systems (cf. e. g. [DJ90]). Due to lack of space and for the sake of readability we restrict our presentation to the one-sorted case (cf. [WGKP93] for the more general many-sorted case). We will consider terms over a signature $sig = (\mathbb{F}, \alpha)$ with sub-signature $cons = (\mathbb{C}, \alpha|_{\mathbb{C}})$ where the functions symbols from $\mathbb{C} \subseteq \mathbb{F}$ are *constructor* and those of $\mathbb{F} \setminus \mathbb{C}$ are *non-constructor* function symbols, and where α denotes the corresponding arity function (all symbols are assumed to have fixed arity). We consider two distinct types of variables corresponding to their intended usage later on. Namely, $\mathrm{V}_{\mathrm{SIG}}$ and $\mathrm{V}_{\mathrm{CONS}}$ denote the set of *general variables* and of *constructor variables*, respectively. The set V of all variables is given by $\mathrm{V} := \mathrm{V}_{\mathrm{SIG}} \uplus \mathrm{V}_{\mathrm{CONS}}$. $\mathcal{T}_{\mathrm{SIG}} := \mathcal{T}(sig, \mathrm{V}_{\mathrm{SIG}} \uplus \mathrm{V}_{\mathrm{CONS}})$ is the set of (variable-mixed) *general terms*, $\mathcal{T}_{\mathrm{CONS}} := \mathcal{T}(cons, \mathrm{V}_{\mathrm{CONS}})$ the set of *pure constructor* terms, and $\mathcal{T}(cons, \mathrm{V}_{\mathrm{SIG}} \uplus \mathrm{V}_{\mathrm{CONS}})$ the set of (variable-mixed) *constructor terms*. $\mathcal{GT}(cons) := \mathcal{T}(cons, \emptyset)$ denotes the set of all *constructor ground terms* and $\mathcal{GT}(sig) := \mathcal{T}(sig, \emptyset)$ the set of all *ground terms*. To avoid problems with empty domains we assume that $\mathcal{GT}(cons)$ is non-empty. In order to avoid confusion note that we have $\mathcal{T}_{\mathrm{CONS}} \subseteq \mathcal{T}_{\mathrm{SIG}}$ but $\mathrm{V}_{\mathrm{CONS}} \cap \mathrm{V}_{\mathrm{SIG}} = \emptyset$. Furthermore, for $\mathrm{X} = \mathrm{X}_{\mathrm{CONS}} \uplus \mathrm{X}_{\mathrm{SIG}}$, $\mathrm{X}_{\mathrm{CONS}} \subseteq \mathrm{V}_{\mathrm{CONS}}$, $\mathrm{X}_{\mathrm{SIG}} \subseteq \mathrm{V}_{\mathrm{SIG}}$, and some set $T = T_{\mathrm{CONS}} \cup T_{\mathrm{SIG}}$ we define $\mathcal{SUB}(\mathrm{X}, T) := \{\sigma : \mathrm{X} \rightarrow T \mid \forall \zeta \in \{\mathrm{SIG}, \mathrm{CONS}\} : \forall x \in \mathrm{X}_\zeta : \sigma(x) \in \mathcal{T}_\zeta\}$. For a term t, we shall use postfix-notation $t\sigma$ instead of $\sigma(t)$.

Instead of the usual sig-algebras we deal with $sig/cons$-algebras with *universe* $\mathcal{A}(\mathrm{SIG})$ and *constructor sub-universe* $\mathcal{A}(\mathrm{CONS}) \subseteq \mathcal{A}(\mathrm{SIG})$ which are defined as usual except for the requirement $c^{\mathcal{A}}[\mathcal{A}(\mathrm{CONS}) \times \ldots \times \mathcal{A}(\mathrm{CONS})] \subseteq \mathcal{A}(\mathrm{CONS})$ (for $c \in \mathbb{C}$). A $sig/cons$-homomorphism $h : \mathcal{A} \longrightarrow \mathcal{B}$ is a usual sig-homomorphism with the additional requirement $h[\mathcal{A}(\mathrm{CONS})] \subseteq \mathcal{B}(\mathrm{CONS})$. For $\mathrm{X} \subseteq \mathrm{V}$ we use $\mathcal{T}(\mathrm{X})$ to denote the term algebra over X and $sig/cons/\mathrm{V}$ (where $\mathcal{T}(\mathrm{X})(\mathrm{SIG}) := \mathcal{T}(sig, \mathrm{X}) \cap \mathcal{T}_{\mathrm{SIG}}$ and $\mathcal{T}(\mathrm{X})(\mathrm{CONS}) := \mathcal{T}(sig, \mathrm{X}) \cap \mathcal{T}_{\mathrm{CONS}}$). For a $sig/cons$-algebra $\mathcal{A}$ and a sig-congruence $\sim$ on $\mathcal{A}(\mathrm{SIG})$, the factor algebra $\mathcal{B}$ of $\mathcal{A}$ modulo $\sim$ (denoted by $\mathcal{A}/{\sim}$) is given by $\mathcal{B}(\mathrm{CONS}) := \{ \sim [\{a\}] \mid a \in \mathcal{A}(\mathrm{CONS}) \}$ with $\mathcal{B}(\mathrm{SIG})$ and $f^{\mathcal{B}}$ as usual (here, $\sim [\{a\}]$ denotes the $\sim$-congruence class of a). For a $sig/cons$-algebra $\mathcal{A}$, an *$\mathcal{A}$-valuation κ of* X is an element of $\mathcal{SUB}(\mathrm{X}, \mathcal{A})$. For DUNNO $\in \{\mathrm{SIG}, \mathrm{CONS}\}$, $dunno \in \{sig, cons\}$ a $sig/cons$-algebra $\mathcal{A}$ is called DUNNO:$dunno$-term-generated if the following holds: $\forall a \in \mathcal{A}(\mathrm{DUNNO}) : \exists t \in \mathcal{GT}(dunno) : \mathcal{A}(t) = a$. $\mathcal{A}$ is called $dunno$-term-generated if it is SIG:$dunno$-term-generated. For K a class of $sig/cons$-algebras, $\mathcal{A}$ a $sig/cons$-algebra, $\mathrm{X} \subseteq \mathrm{V}$ and $\kappa \in \mathcal{SUB}(\mathrm{X}, \mathcal{A})$ we need the following definitions: $\mathcal{A}$ is *initial in* K if $\mathcal{A} \in K$ and for all $\mathcal{B} \in K$ there is a unique $h : \mathcal{A} \to \mathcal{B}$. $\mathcal{A}$ is *free for* K *over* X *w. r. t.* κ if for all $\mathcal{B} \in K$ and $\mu \in \mathcal{SUB}(\mathrm{X}, \mathcal{B})$ there is a unique $h : \mathcal{A} \to \mathcal{B}$ with $\mu = \kappa h$. $\mathcal{A}$ is *free in* K *over* X *w. r. t.* κ if $\mathcal{A} \in K$ and $\mathcal{A}$ is free for K over X w. r. t. κ.

Note that our notion of *sig/cons*-algebras can be viewed as a very special (and simple) case of order-sorted structures, e. g., in the sense of [SNGM89].

2.2 Constructor-Based Positive/Negative Conditional Equational Specifications

Definition 1. (Positive/Negative Conditional Term Rewriting System)
A *positive/negative conditional term rewriting system* (*PNCTRS* for short) over *sig/cons*/V is a set of rules[2] of the form

$$\bigwedge_{i=1}^{m}(s_i = t_i) \wedge \bigwedge_{j=1}^{n}(u_j \neq v_j) \wedge \bigwedge_{k=1}^{p}(\mathrm{Def}\, w_k) \quad \Rightarrow \quad l = r$$

with all s_i, t_i, u_j, v_j, w_k in $\mathcal{T}(sig, \mathrm{V}_{\mathrm{SIG}} \uplus \mathrm{V}_{\mathrm{CONS}})$.

Definition 2. (Semantics of PNCTRSs)
If R is a PNCTRS over *sig/cons*/V then a *sig/cons*-algebra $\mathcal{A}$ is said to be a *model of* R if we have:

$\forall (P \Rightarrow l{=}r) \in R$: $\forall \kappa \in \mathcal{SUB}(\mathrm{V}, \mathcal{A})$: $((P$ is true w.r.t. $\mathcal{A}_\kappa) \Rightarrow \mathcal{A}_\kappa(l) = \mathcal{A}_\kappa(r))$

where P is true w. r. t. $\mathcal{A}_\kappa$ if all its literals are true w. r. t. $\mathcal{A}_\kappa$, i. e., $\mathcal{A}_\kappa(u) = \mathcal{A}_\kappa(v)$ for $(u = v)$ in P, $\mathcal{A}_\kappa(u) \neq \mathcal{A}_\kappa(v)$ for $(u \neq v)$ in P, and $\mathcal{A}_\kappa(u) \in \mathcal{A}(\mathrm{CONS})$ for $(\mathrm{Def}\, u)$ in P.

Definition 3. (Minimal and Minimum Model)
Let the quasi-orderings $\precsim_{\mathrm{H}}$ and $\precsim_{\mathrm{CONS}}$ on *sig/cons*-algebras be defined by: $\mathcal{A} \precsim_{\mathrm{H}} \mathcal{B}$ if there exists a *sig/cons*-homomorphism from $\mathcal{A}$ to $\mathcal{B}$. $\mathcal{A} \precsim_{\mathrm{CONS}} \mathcal{B}$ if there exists a *cons*-homomorphism from $\mathcal{A}|_{\mathbb{C} \uplus \{\mathrm{CONS}\}}$ to $\mathcal{B}|_{\mathbb{C} \uplus \{\mathrm{CONS}\}}$.[3]
A *sig/cons*-algebra $\mathcal{A}$ is a *minimal model* of a PNCTRS R if it is minimal w. r. t. $\precsim_{\mathrm{H}}$ in the class M of all models of R. It is a *constructor-minimal model* of R if it is minimal w. r. t. $\precsim_{\mathrm{CONS}}$ in M. $\mathcal{A}$ is a *minimum model* of R if it is a least model of R w. r. t. $\precsim_{\mathrm{H}}$, and it is a *constructor-minimum model* of R if it is a least model of R w. r. t. $\precsim_{\mathrm{CONS}}$.

Now, every PNCTRS possesses minimal models, but not necessarily a minimum model. But as we shall see, under certain restrictions we are able to guarantee the existence of a constructor-minimum model which can be constructed as a quotient term algebra. To this end we have to define a reduction relation for PNCTRSs. This is only possible in a reasonable way — without assuming some termination ordering conditions as in [Kap87] — by imposing additional restrictions on the structure of the rules. To be precise, we require for every *constructor rule* of R, i. e., for every $(P \Rightarrow l{=}r) \in R$ with $l \in \mathcal{T}(cons, \mathrm{V}_{\mathrm{SIG}} \uplus \mathrm{V}_{\mathrm{CONS}})$:

---

[2] We shall always use '$=$' in equations or rules. The interpretation as '$=$' or '$\longrightarrow$' tacitly depends on the context.

[3] By the notation $\mathcal{A}|_{\mathbb{C} \uplus \{\mathrm{CONS}\}}$ we mean the *cons*-algebra of $\mathcal{A}$ with universe $\mathcal{A}(\mathrm{CONS})$.

(1) There are no negative equations in P (*constructor rules have "Horn"-form*), and
(2) All terms in r and P are constructor terms, i. e., in $\mathcal{T}(cons, \mathrm{V}_{\mathrm{SIG}} \uplus \mathrm{V}_{\mathrm{CONS}})$, and all variables of r and P occur in l (*"constructor-preservation"*).

PNCTRSs satisfying these conditions are called *constructor-based*. In the rest of the paper all PNCTRSs are tacitly assumed to be constructor-based.

Now we are prepared to define the reduction relation.

Definition 4. (Reduction Relation)
Let R be a (constructor-based) PNCTRS over $sig/cons/\mathrm{V}$ and $R_{\mathbf{C}}$ its subset of constructor rules.[4] For $\mathrm{X} \subseteq \mathrm{V}$ the *reduction relation* $\longrightarrow_{R,\mathrm{X}}$ *on* $\mathcal{T}(sig, \mathrm{X})$ induced by R is the smallest relation $\longrightarrow$ satisfying $\longrightarrow \cap (\mathcal{GT}(cons) \times \mathcal{T}_{\mathrm{SIG}}) \subseteq \longrightarrow_{R_{\mathbf{C}},\mathrm{X}}$,[5] and $s \longrightarrow t$ if $s, t \in \mathcal{T}(sig, \mathrm{X})$ and

$$\exists (P \Rightarrow l{=}r) \in R\colon\ \exists \sigma \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}(\mathrm{X}))\colon\ \exists p \in \mathcal{POS}(s)\colon \begin{pmatrix} s/p = l\sigma \wedge \\ t = s[p \leftarrow r\sigma] \wedge \\ P\sigma \text{ fulfilled w.r.t. } \longrightarrow \end{pmatrix}$$

where "Q is fulfilled w. r. t. $\longrightarrow$" is a shorthand for

$$\begin{pmatrix} \quad \forall (u = v) \text{ in } Q : u \downarrow v \\ \wedge\ \forall (\mathrm{Def}\, u) \text{ in } Q : \exists \hat{u} \in \mathcal{GT}(cons) : u \longrightarrow^{*} \hat{u} \\ \wedge\ \forall (u \neq v) \text{ in } Q : \exists \hat{u}, \hat{v} \in \mathcal{GT}(cons) : u \longrightarrow^{*} \hat{u} \not\downarrow \hat{v} \longleftarrow^{*} v \end{pmatrix}$$

The well-definedness of $\longrightarrow_{R,\mathrm{X}}$ can be established by a double closure construction, firstly for the constructor rules only (which is possible due to their required Horn-form), and secondly for general rules knowing that the reduction relation on constructor ground terms remains invariant (due to the required constructor-preservation property).

Note moreover that $\longrightarrow_{R,\mathrm{X}}$ is stable under substitutions from $\mathcal{SUB}(\mathrm{V}, \mathcal{T}(\mathrm{X}))$. Furthermore, for $\mathrm{X} \subseteq \mathrm{Y}$, confluence of $\longrightarrow_{R,\mathrm{Y}}$ implies confluence of $\longrightarrow_{R,\mathrm{X}}$.

In order to guarantee the existence of a constructor-minimum model of R we have to require that the reduction relation is (ground) confluent[6] and that the terms of the negative equations in the conditions of R are "defined".

Definition 5. (Def-Moderate PNCTRS)
A (constructor-based) PNCTRS is called Def-*moderate* (Def-MCTRS for short) if for each rule $(P \Rightarrow l{=}r) \in R$ and each negative condition $(u \neq v)$ in P we have that $(\mathrm{Def}\, u)$, $(\mathrm{Def}\, v)$ are in P, too.

Theorem 6. *(Existence and Characterization of a Constructor-Minimum Model) Let R be a* Def-*MCTRS over $sig/cons/\mathrm{V}$ and K be the class of all constructor-minimal models of R. Moreover, let $\mathrm{X} \subseteq \mathrm{V}$ and κ be given by: $x \mapsto\ \longleftrightarrow^{*}_{R,\mathrm{X}} [\{x\}]$. Then the following holds:*

---

[4] i. e., $R_{\mathbf{C}}$ is a positive conditional system consisting of those rules of R that do not involve non-constructor function symbols.

[5] This requirement ensures minimality of $\longrightarrow$ on the constructor (ground) terms.

[6] Although many of our results do not depend on the (ground) confluence assumption, this property is desirable anyway, since otherwise, for instance the congruence induced by the reduction relation of some R need not yield a model of R in general.

- *If $\longrightarrow_{R,\emptyset}$ is confluent then $\mathcal{T}(\mathrm{X})/\longleftrightarrow^*_{R,\mathrm{X}}$ is free for K over X w. r. t. κ.*
- *If $\longrightarrow_{R,\mathrm{X}}$ is confluent and $\mathrm{X} \subseteq \mathrm{V}_{\mathrm{SIG}}$, then $\mathcal{T}(\mathrm{X})/\longleftrightarrow^*_{R,\mathrm{X}}$ is a minimal model of R which is free in K over X w. r. t. κ and which is moreover a constructor-minimum model of R.*

Corollary 7. *If R is a* Def-*MCTRS such that $\longrightarrow_{R,\emptyset}$ is confluent then $\mathcal{GT}/\longleftrightarrow^*_{R,\emptyset}$ is a minimal model of R which is initial in the class of all constructor-minimal models of R and which is the (up to isomorphism) unique $\precsim_{\mathrm{H}}$-minimum of all sig-term-generated constructor-minimal models of R.*

Hence, for confluent $\longrightarrow_{R,\emptyset}$ the factor algebra $\mathcal{GT}/\longleftrightarrow^*_{R,\emptyset}$ provides us with a constructive operational characterization of the intended unique minimal model. Moreover, the reduction relation is monotonic w. r. t. consistent extension in the following sense.

Theorem 8. *(Monotonicity of $\longrightarrow_{R,\mathrm{X}}$ w. r. t. Consistent Extension)*
Let R be a PNCTRS over $sig/cons/\mathrm{V}$ and let R' be another PNCTRS over $sig'/cons'/\mathrm{V}'$ with $\left| \begin{array}{c} sig' = (\mathbb{F}', \alpha') \\ cons' = (\mathbb{C}', \alpha'|_{\mathbb{C}'}) \\ \mathrm{V}' = \mathrm{V} \end{array} \right| \begin{array}{c} \mathbb{F} \subseteq \mathbb{F}' \\ \mathbb{C} = \mathbb{C}' \\ \alpha \subseteq \alpha' \end{array} \left| \begin{array}{c} R \subseteq R' \\ \mathrm{X} \subseteq \mathrm{X}' \\ \end{array} \right|.$
Moreover assume that no new (i. e., of a rule from $R' \setminus R$) left-hand side is a constructor term. Then the following properties hold:

- *The reduction relations $\longrightarrow_{R,\mathrm{X}}$ and $\longrightarrow_{R',\mathrm{X}'}$ coincide for constructor terms, i. e.: $\forall s \in \mathcal{T}(cons, \mathrm{X}) : \forall t : (s \longrightarrow^*_{R',\mathrm{X}'} t \;\Leftrightarrow\; s \longrightarrow^*_{R,\mathrm{X}} t \;\Leftrightarrow\; s \longrightarrow^*_{R_{\mathbb{C}},\mathrm{X}} t)$.*
- *The reduction relation $\longrightarrow_{R,\mathrm{X}}$ is monotonic in X and R: $\longrightarrow_{R,\mathrm{X}} \subseteq \longrightarrow_{R',\mathrm{X}'}$.*

In [WG93] we have developed a couple of confluence criteria for PNCTRSs as well as some slightly extended "decreasingness"-notions which generalize known results for positive conditional rewrite systems (e. g., of [DOS88]). For related work on PNCTRSs see also [AB92], [Bec93].

3 Inductive Validity: Notions and Interrelations

Definition 9. (Syntax of Formulas)
Let $\mathrm{X} \subseteq \mathrm{V}$. The set of *formulas* (or *Gentzen clauses*) over sig, X is defined to be $\mathrm{FORM}(sig, \mathrm{X}) := \mathrm{ATOM}(sig, \mathrm{X})^* \times \mathrm{ATOM}(sig, \mathrm{X})^*$, where $\mathrm{ATOM}(sig, \mathrm{X})$ is the set of *atoms* over the predicate symbols '=' and 'Def' on terms from $\mathcal{T}(sig, \mathrm{X})$. A formula (Γ, Δ) will also be denoted by $\Gamma \longrightarrow \Delta$. For the special cases of $\longrightarrow \Delta$ and $(s = t) \longrightarrow$ we also write Δ and $(s \neq t)$, respectively.

Definition 10. (Validity of Formulas in Algebras)
Let $\mathrm{X} \subseteq \mathrm{V}$, $\mathcal{A}$ be a *sig/cons*-algebra, and $\kappa \in \mathcal{SUB}(\mathrm{X}, \mathcal{A})$.
An atom $(u = v) \in \mathrm{ATOM}(sig, \mathrm{X})$ is *true w. r. t. $\mathcal{A}_\kappa$* if $\mathcal{A}_\kappa(u) = \mathcal{A}_\kappa(v)$.
An atom $(\mathrm{Def}\, u) \in \mathrm{ATOM}(sig, \mathrm{X})$ is *true w. r. t. $\mathcal{A}_\kappa$* if $\mathcal{A}_\kappa(u) \in \mathcal{A}(\mathrm{CONS})$.
A formula $(\Gamma \longrightarrow \Delta) \in \mathrm{FORM}(sig, \mathrm{X})$ is *valid in $\mathcal{A}$* if $\forall \kappa \in \mathcal{SUB}(\mathrm{X}, \mathcal{A})$:
$(\forall A$ in Γ: (A is true w. r. t. $\mathcal{A}_\kappa$) $\;\Rightarrow\;$ $\exists B$ in Δ: (B is true w. r. t. $\mathcal{A}_\kappa$)).

Next we introduce a notion for substitutions that replace constructor variables by constructor ground terms and leave the general variables invariant.

Definition 11. (Inductive Substitutions)
The set of *inductive substitutions* is defined by: INDSUB(V, *cons*)
$:= \{\tau \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}(\mathrm{V_{SIG}})) \mid \tau[\mathrm{V_{CONS}}] \subseteq \mathcal{GT}(cons) \wedge \tau|_{\mathrm{V_{SIG}}}{=}\mathrm{id}|_{\mathrm{V_{SIG}}}\}$.

Definition 12. (Type-A / B' / B / C / D' / D / E Inductive Validity)
Let R be a PNCTRS over $sig/cons/\mathrm{V}$, let M be the class of all models of R and K be the class of all constructor-minimal models of R. Then a formula $(\Gamma \longrightarrow \Delta) \in \mathrm{FORM}(sig, \mathrm{V})$ is said to be

— *type-A inductively valid w. r. t. R*, denoted by $R \models^{A}_{ind} \Gamma \longrightarrow \Delta$, if
  $\forall \mathcal{A} \in M$: $\forall \tau \in \mathrm{INDSUB}(\mathrm{V}, cons)$: $(\Gamma \longrightarrow \Delta)\tau$ is valid in $\mathcal{A}$.
— *type-B′ inductively valid w. r. t. R*, denoted by $R \models^{B'}_{ind} \Gamma \longrightarrow \Delta$, if
  $\forall \mathcal{A} \in M$: (($\mathcal{A}$ is CONS:*cons*-term-generated) $\Rightarrow$ $(\Gamma \longrightarrow \Delta)$ is valid in $\mathcal{A}$).
— *type-B inductively valid w. r. t. R*, denoted by $R \models^{B}_{ind} \Gamma \longrightarrow \Delta$, if
  $\forall \mathcal{A} \in K$: $\forall \tau \in \mathrm{INDSUB}(\mathrm{V}, cons)$: $(\Gamma \longrightarrow \Delta)\tau$ is valid in $\mathcal{A}$.
— *type-C inductively valid w. r. t. R*, denoted by $R \models^{C}_{ind} \Gamma \longrightarrow \Delta$, if
  $\forall \mathcal{A} \in K$: (($\mathcal{A}$ is CONS:*cons*-term-generated) $\Rightarrow$ $(\Gamma \longrightarrow \Delta)$ is valid in $\mathcal{A}$).
— *type-D′ inductively valid w. r. t. R*, denoted by $R \models^{D'}_{ind} \Gamma \longrightarrow \Delta$, if
  $\forall \mathcal{A} \in K$: (($\mathcal{A}$ is SIG:*cons*-term-generated) $\Rightarrow$ $(\Gamma \longrightarrow \Delta)$ is valid in $\mathcal{A}$).
— *type-D inductively valid w. r. t. R*, denoted by $R \models^{D}_{ind} \Gamma \longrightarrow \Delta$, if
  $(\Gamma \longrightarrow \Delta)$ is valid in $\mathcal{T}(\mathrm{V_{SIG}})/\overset{*}{\leftrightarrow}_{R,\mathrm{V_{SIG}}}$.
— *type-E inductively valid w. r. t. R*, denoted by $R \models^{E}_{ind} \Gamma \longrightarrow \Delta$, if
  $(\Gamma \longrightarrow \Delta)$ is valid in $\mathcal{GT}/\overset{*}{\leftrightarrow}_{R,\emptyset}$.

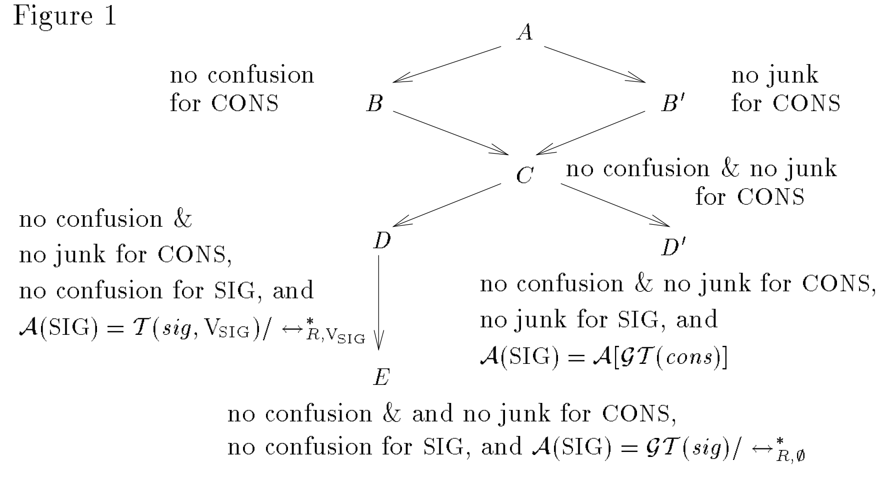
Type-A formulates the idea that (constructor) variables are meant to denote objects denoted by (constructor) ground terms. However, contrary to all other types, type-A does not restrict the models of the specification that have to be considered, but considers only instances of formulas obtained by inductive substitutions. Type-B' forbids junk in the constructor sub-universe (i. e. the domain of interest), which makes inductive substitutions redundant. Type-B forbids unnecessary confusion in the constructor sub-universe. Type-C combines both restrictions, which is appealing if one wants to prescribe a precise and fixed knowledge on the basic objects for computation. Type-D' corresponds to the philosophy that a partially defined function is to be interpreted as the set of all possible complete and consistent extensions. Type-D and E finally fix one specific unique minimal model which has neither junk nor confusion in the constructor sub-universe and which can be described constructively in terms of the factor algebra of the term algebra modulo the congruence induced by the reduction relation (provided the latter is confluent). Type-E uses the ground term algebra $\mathcal{GT}$, which is only adequate (cf. Theorem 17 and Example 14) when no general variables occur in the formula. Therefore, Type-D uses the term algebra $\mathcal{T}(\mathrm{V_{SIG}})$ over $\mathrm{V_{SIG}}$. In a sense, type-D means *inductive semantics* for $\mathrm{V_{CONS}}$ and *free semantics* for $\mathrm{V_{SIG}}$. The notations of the types of inductive validity are motivated by the following result:

Lemma 13. *(From Type-A down to Type-E)*
Let R be a PNCTRS over $sig/cons/\mathrm{V}$ and $\Gamma \longrightarrow \Delta \in \mathrm{FORM}(sig, \mathrm{V})$.

(a) If $R \models^{A}_{ind} \Gamma \longrightarrow \Delta$ *, then* $R \models^{B}_{ind} \Gamma \longrightarrow \Delta$ *and* $R \models^{B'}_{ind} \Gamma \longrightarrow \Delta$.
(b) If $R \models^{B}_{ind} \Gamma \longrightarrow \Delta$ *or* $R \models^{B'}_{ind} \Gamma \longrightarrow \Delta$ *, then* $R \models^{C}_{ind} \Gamma \longrightarrow \Delta$.
(c) If $R \models^{C}_{ind} \Gamma \longrightarrow \Delta$ *, then* $R \models^{D'}_{ind} \Gamma \longrightarrow \Delta$ *and even* $R \models^{D'}_{ind} \Gamma' \longrightarrow \Delta$ *where Γ' results form Γ by deleting (some)* Def*-atoms.*
(d) If R is a Def*-MCTRS, $\rightarrow_{R,\mathrm{V_{SIG}}}$ is confluent, and* $R \models^{C}_{ind} \Gamma \longrightarrow \Delta$ *, then* $R \models^{D}_{ind} \Gamma \longrightarrow \Delta$.
(e) If $R \models^{D}_{ind} \Gamma \longrightarrow \Delta$ *, then* $R \models^{E}_{ind} \Gamma \longrightarrow \Delta$.
(f) If R is a Def*-MCTRS, $\rightarrow_{R,\emptyset}$ is confluent, and* $R \models^{C}_{ind} \Gamma \longrightarrow \Delta$ *, then* $R \models^{E}_{ind} \Gamma \longrightarrow \Delta$.
(g) Under the restrictive condition of sufficient completeness of $\rightarrow_{R,\emptyset}$ (i. e.: $\forall s \in \mathcal{GT}(sig)$: $\exists t \in \mathcal{GT}(cons)$: $s \rightarrow^{*}_{R,\emptyset} t$) the following holds:
If R is a Def*-MCTRS and $\rightarrow_{R,\emptyset}$ is confluent, then we get:*
$R \models^{D'}_{ind} \Gamma \longrightarrow \Delta \quad \Leftrightarrow \quad R \models^{E}_{ind} \Gamma \longrightarrow \Delta$.

The basic characteristics of and the relations between these notions of inductive validity are illustrated in Figure 1 below where implications are indicated by arrows. Missing arrows indicate non-implications as shown by Example 14 below.

Figure 1



Example 14. *Let us return to the specification on natural numbers from section 1 and start with $\mathbb{F} := \mathbb{C} := \{s, 0\}$ and $R := \emptyset$. Then we have*

$$R \models^{B}_{ind} s(0) \neq 0 \text{ , but } R \not\models^{A}_{ind} s(0) \neq 0 \text{ ,}$$

since 0 and $s(0)$ are interpreted as distinct objects in any constructor-minimal model of R, and since $s(0) \neq 0$ does not hold for instance in the trivial model of

R which identifies everything. Furthermore, for the same reason we have

$$R \models^{C}_{ind} s(0) \neq 0 \ , \ but \ \ R \not\models^{B'}_{ind} s(0) \neq 0 \ .$$

Keeping $\mathbb{C}' := \mathbb{C}$, *let us add two non-constructor symbols* $+$ *and* ω, *yielding* $\mathbb{F}' := \{+, \omega\} \uplus \mathbb{C}$, *with the rules*

$$R'\colon \ 0 + y = y, \quad s(x) + y = s(x + y),$$

where it does not matter whether x, y *are from* $\mathrm{V_{SIG}}$ *or* $\mathrm{V_{CONS}}$. *Then we have*

$$R' \models^{C}_{ind} (\,\mathrm{Def}\,\omega\,) \longrightarrow (\,\omega{+}0 = \omega\,) \ , \ but \ \ R' \not\models^{B}_{ind} (\,\mathrm{Def}\,\omega\,) \longrightarrow (\,\omega{+}0 = \omega\,) \ ,$$

since if $(\mathrm{Def}\,\omega)$ *is valid in some* CONS:*cons-term-generated model then* ω *can be denoted by some constructor ground term* $s^i(0)$, *and since for type-B inductive validity we also have to consider (constructor-minimal) models containing constructor objects that are not denoted by some term of the form* $s^i(0)$, *e. g.* $\mathcal{GT}'/\leftrightarrow^{*}_{R',\emptyset}$ *where* $\mathcal{GT}'$ *differs from* $\mathcal{GT}$ *only in the constructor sub-universe given as* $\mathcal{GT}'(\mathrm{CONS}) := \mathcal{GT}(\mathrm{SIG})$. *Note that* $\mathcal{GT}'/\leftrightarrow^{*}_{R',\emptyset}$ *is constructor-minimal indeed, due to* $\mathcal{GT}'/\leftrightarrow^{*}_{R',\emptyset} \precsim_{\mathrm{CONS}} \mathcal{GT}/\leftrightarrow^{*}_{R',\emptyset}$. *Furthermore, for the same reason we have*

$$R' \models^{B'}_{ind} (\,\mathrm{Def}\,\omega\,) \longrightarrow (\,\omega{+}0 = \omega\,) \ , \ but \ \ R' \not\models^{A}_{ind} (\,\mathrm{Def}\,\omega\,) \longrightarrow (\,\omega{+}0 = \omega\,) \ .$$

Again keeping $\mathbb{C}'' := \mathbb{C}$, *let us add a non-constructor symbol* $-$, *yielding* $\mathbb{F}'' := \{-\} \uplus \mathbb{C}$, *with the rules*

$$R''\colon \ x - 0 = x, \quad s(x) - s(y) = x - y,$$

where it does not matter whether x, y *are from* $\mathrm{V_{SIG}}$ *or* $\mathrm{V_{CONS}}$. *Then we have*

$$R'' \models^{D}_{ind} \ 0 - s(0) \neq 0 \ , \ but \ \ R'' \not\models^{D'}_{ind} \ 0 - s(0) \neq 0$$

because $0 - s(0) \neq 0$ *does not hold in the* SIG:*cons-term-generated constructor-minimal model obtained by identifying* $0 - y$ *with* 0. *Furthermore,*

$$R'' \models^{D'}_{ind} \ (0 - s(0)) - (0 - s(0)) = 0 \ , \ but \ \ R'' \not\models^{E}_{ind} \ (0 - s(0)) - (0 - s(0)) = 0$$

because a SIG:*cons-term-generated model must satisfy* $0 - s(0) = s^i(0)$ *for some* i, *but* $(0{-}s(0)) - (0{-}s(0)) \not\leftrightarrow^{*}_{R'',\emptyset} 0$ *(i. e.,* $(0{-}s(0)) - (0{-}s(0)) = 0$ *does not hold in the* CONS:*cons-term-generated, constructor-minimal model* $\mathcal{GT}/\leftrightarrow^{*}_{R'',\emptyset}$ *of* R''*).*

Finally, keeping $\mathbb{C}''' := \mathbb{C}$ *and choosing* $\mathbb{F}''' := \{+\} \uplus \mathbb{C}$, *for* $X \in \mathrm{V_{SIG}}$ *we have* $R' \models^{E}_{ind} X{+}0 = X$ *(which is not the case for the consistent extension via* $\mathbb{F}''' := \mathbb{F}'$*), but* $R' \not\models^{D}_{ind} X{+}0 = X$, *since* $\forall t \in \mathcal{GT}(sig)\colon\ t + 0 \leftrightarrow^{*}_{R',\emptyset} t$ *but not* $X{+}0 \leftrightarrow^{*}_{R',\mathrm{V_{SIG}}} X$.

While the example has shown that the reverse of each implication depicted in the figure does not hold in general, the following lemma gives sufficient conditions.

Lemma 15. *(From Type-E up to Type-A)*
Let R *be a PNCTRS over* $sig/cons/\mathrm{V}$ *and* $\Gamma \longrightarrow \Delta \in \mathrm{FORM}(sig, \mathrm{V})$.

(a) If $\Gamma \longrightarrow \Delta$ *does not contain variables from* $\mathrm{V_{SIG}}$, *then we get:*
$$R \models^{E}_{ind} \Gamma \longrightarrow \Delta \quad \Rightarrow \quad R \models^{D}_{ind} \Gamma \longrightarrow \Delta \quad .$$

(b) If R *is a* Def-*MCTRS,* $\rightarrow_{R,\emptyset}$ *is confluent, and if for all (top level) terms* u *of atoms in* Γ *we have* $R \models^{D}_{ind} (\mathrm{Def}\,u)$, *then we get:*
$$R \models^{D}_{ind} \Gamma \longrightarrow \Delta \quad \Rightarrow \quad R \models^{C}_{ind} \Gamma \longrightarrow \Delta \quad .$$

(c) If for each atom $(\mathrm{Def}\, u)$ *in* Γ *we have* $R \models^{C}_{ind} (\mathrm{Def}\, u)$,[7] *then we get:*
$$R \models^{C}_{ind} \Gamma \longrightarrow \Delta \quad \Rightarrow \quad R \models^{B}_{ind} \Gamma \longrightarrow \Delta \quad .$$
If for each atom $(\mathrm{Def}\, u)$ *in* Γ *we have* $R \models^{B'}_{ind} (\mathrm{Def}\, u)$,[7] *then we get:*
$$R \models^{B'}_{ind} \Gamma \longrightarrow \Delta \quad \Rightarrow \quad R \models^{A}_{ind} \Gamma \longrightarrow \Delta \quad .$$
(d) If no rule in R *has a negative condition (like* $u \neq v$*), then we get:*
$$R \models^{C}_{ind} \Delta \quad \Rightarrow \quad R \models^{A}_{ind} \Delta \quad .$$

Note that (by Lemma 13) Lemma 15 also permits to conclude from type-C to B' (via A), from type-B to A (via C), and from type-D' to C (via E, D).

Lemma 16. *(Operational Characterization of Type-D Inductive Validity)*
Let R *be a PNCTRS over* $sig/cons/\mathrm{V}$ *and* $\Gamma \longrightarrow \Delta \in \mathrm{FORM}(sig, \mathrm{V})$.
Then $R \models^{D}_{ind} \Gamma \longrightarrow \Delta$ *is equivalent to* $\forall \tau \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}(\mathrm{V_{SIG}}))$:
$$\begin{pmatrix} \forall (u{=}v) \text{ in } \Gamma\text{: } u\tau \leftrightarrow^{*}_{R,\mathrm{V_{SIG}}} v\tau \ \wedge\ \forall (\mathrm{Def}\, u) \text{ in } \Gamma\text{: } \exists \hat{u} \in \mathcal{GT}(cons)\text{: } u\tau \leftrightarrow^{*}_{R,\mathrm{V_{SIG}}} \hat{u} \\ \Rightarrow \\ \exists (u{=}v) \text{ in } \Delta\text{: } u\tau \leftrightarrow^{*}_{R,\mathrm{V_{SIG}}} v\tau \ \vee\ \exists (\mathrm{Def}\, u) \text{ in } \Delta\text{: } \exists \hat{u} \in \mathcal{GT}(cons)\text{: } u\tau \leftrightarrow^{*}_{R,\mathrm{V_{SIG}}} \hat{u} \end{pmatrix}$$

Finally we show that (under some reasonable assumptions) all defined notions of inductive validity are monotonic w. r. t. consistent extension.

Theorem 17. *(Monotonicity of Inductive Validity w. r. t. Consistent Extension)*
Let R *be a PNCTRS over* $sig/cons/\mathrm{V}$ *and let* R' *be another PNCTRS over* $sig'/cons'/\mathrm{V}'$ *with*[8]
$$\left|\begin{array}{c} sig' = (\mathbb{F}', \alpha') \\ cons' = (\mathbb{C}', \alpha'|_{\mathbb{C}'}) \\ \mathrm{V}' = \mathrm{V} \end{array}\right|\begin{array}{c} \mathbb{F} \subseteq \mathbb{F}' \\ \mathbb{C} = \mathbb{C}' \\ \alpha \subseteq \alpha' \end{array}\left|\begin{array}{c} R \subseteq R' \\ \mathrm{X} \subseteq \mathrm{X}' \\ \ \end{array}\right| .$$
Moreover assume that
$$(*) \quad \rightarrow_{R',\emptyset} \text{ is confluent}$$
and that the following condition for the new left hand sides holds:
$$(**) \quad \forall\, (C \Rightarrow l{=}r) \in R' \setminus R\text{: } \ l \notin \mathcal{T}(cons, \mathrm{V_{SIG}} \uplus \mathrm{V_{CONS}}).$$
Then, for any clause $\Gamma \longrightarrow \Delta \in \mathrm{FORM}(sig, \mathrm{V})$ *we have:*

(A) $R \models^{A}_{ind} \Gamma \longrightarrow \Delta \ \Rightarrow\ R' \models^{A}_{ind} \Gamma \longrightarrow \Delta$ *(even without assuming* $(*)$, $(**)$*).*
(B′) $R \models^{B'}_{ind} \Gamma \longrightarrow \Delta \ \Rightarrow\ R' \models^{B'}_{ind} \Gamma \longrightarrow \Delta$ *(even without assuming* $(*)$, $(**)$*).*
(B) If R' *is* Def*-moderate then we get:* $R \models^{B}_{ind} \Gamma \longrightarrow \Delta \ \Rightarrow\ R' \models^{B}_{ind} \Gamma \longrightarrow \Delta$.
(C) If R' *is* Def*-moderate then we get:* $R \models^{C}_{ind} \Gamma \longrightarrow \Delta \ \Rightarrow\ R' \models^{C}_{ind} \Gamma \longrightarrow \Delta$.
(D′) If R' *is* Def*-moderate then we get:* $R \models^{D'}_{ind} \Gamma \longrightarrow \Delta \ \Rightarrow\ R' \models^{D'}_{ind} \Gamma \longrightarrow \Delta$.
(D) If for all (top level) terms u *of atoms in* Γ *we have* $R \models^{D}_{ind} (\mathrm{Def}\, u)$,
then we get: $R \models^{D}_{ind} \Gamma \longrightarrow \Delta \ \Rightarrow\ R' \models^{D}_{ind} \Gamma \longrightarrow \Delta$.
(E) If $\Gamma \longrightarrow \Delta$ *does not contain variables from* $\mathrm{V_{SIG}}$,
and if for all (top level) terms u *of atoms in* Γ *we have* $R \models^{E}_{ind} (\mathrm{Def}\, u)$,
then we get: $R \models^{E}_{ind} \Gamma \longrightarrow \Delta \ \Rightarrow\ R' \models^{E}_{ind} \Gamma \longrightarrow \Delta$.

---

[7] Even if this condition is not satisfied, the following equivalence transformation for type-B', C, D', D, and E validity (but not for A and B) may help to apply the Lemma : $\Gamma, (\mathrm{Def}\, u), \Gamma' \longrightarrow \Delta$ is equivalent to $\Gamma, (x{=}u), \Gamma' \longrightarrow \Delta$ for a fresh (i. e. not occurring in Γ, u, Γ', Δ) constructor variable x.

[8] In a sorted framework one may even add new constructor symbols for new sorts and permit new rules with left-hand sides that are new (but not old) constructor terms.

Proof. Let $\mathcal{A}'$ be some $sig'/cons'$-algebra. We define the $sig/cons$-algebra $\mathcal{A}$ by $\mathcal{A} := \mathcal{A}'|_{\mathbb{F}\uplus\{\mathrm{SIG},\mathrm{CONS}\}}$. Now an "old" formula $\Gamma \longrightarrow \Delta \in \mathrm{FORM}(sig, \mathrm{V})$ is valid in $\mathcal{A}$ if and only if it is valid in $\mathcal{A}'$. Its inductive instances $(\Gamma \longrightarrow \Delta)\tau$ do not differ for $\tau \in \mathrm{INDSUB}(\mathrm{V}, cons)$ and $\tau \in \mathrm{INDSUB}(\mathrm{V}', cons')$ due to $\mathcal{GT}(cons) = \mathcal{GT}(cons')$. Furthermore, if $\mathcal{A}'$ is CONS:$cons'$-term-generated (or even SIG:$cons'$-term-generated), then $\mathcal{A}$ is CONS:$cons$-term-generated (or even SIG:$cons$-term-generated). Thus for proving (A), $(B'\ )$, [and (B), (C), (D'),] it suffices to show the following claim: $(***)$ If $\mathcal{A}'$ is a [constructor-minimal] model of R', then $\mathcal{A}$ is a [constructor-minimal] model of R.

Proof of $(***)$*:* Assume that $\mathcal{A}'$ is a model of R'. Then $\mathcal{A}$, as defined above, is a model of R since each rule from R can be translated into an "old" formula (on which $\mathcal{A}$ and $\mathcal{A}'$ do not differ). Let us now assume that R' is Def-moderate and that $\mathcal{A}'$ is constructor-minimal. Let $\mathcal{GT}'$ and $\mathcal{GT}(cons)$ denote the ground term algebra over $sig'/cons'$ and $cons$, respectively. By $(*)$ and Theorem 6, $\mathcal{GT}'/\leftrightarrow^*_{R',\emptyset}$ is a constructor-minimum model of R' w. r. t. $sig'/cons'$. Thus, $\mathcal{A}'$ must be a constructor-minimum model of R' w. r. t. $sig'/cons'$, too. Hence, there exists some $cons'$-homomorphism $h' : \mathcal{A}'|_{\mathbb{C}'\uplus\{\mathrm{CONS}\}} \rightarrow (\mathcal{GT}'/\leftrightarrow^*_{R',\emptyset})|_{\mathbb{C}'\uplus\{\mathrm{CONS}\}}$. By defining $h(a) := h'(a) \cap \mathcal{GT}(cons)$ (for $a \in \mathcal{A}(\mathrm{CONS})$), we get a $cons$-homomorphism $h : \mathcal{A}|_{\mathbb{C}\uplus\{\mathrm{CONS}\}} \rightarrow \mathcal{GT}(cons)/(\leftrightarrow^*_{R',\emptyset} \cap (\mathcal{GT}(cons)\times\mathcal{GT}(cons)))$. By confluence of $\longrightarrow_{R',\emptyset}$ and Theorem 8 we get $(\leftrightarrow^*_{R',\emptyset} \cap (\mathcal{GT}(cons)\times\mathcal{GT}(cons)))$ $\subseteq (\downarrow_{R',\emptyset} \cap (\mathcal{GT}(cons)\times\mathcal{GT}(cons))) \subseteq \downarrow_{R_{\mathbb{C}},\emptyset} \subseteq \leftrightarrow^*_{R_{\mathbb{C}},\emptyset} \subseteq \ker(\mathcal{B})$ for the kernel of any model $\mathcal{B}$ of R, and by the Homomorphism-Theorem we then get $\mathcal{A} \precsim_{\mathrm{CONS}} \mathcal{B}$. Thus, $\mathcal{A}$ is not only a model of R, but also a constructor-minimal one. ∎

Proof of (D): We use the operational version of type-D inductive validity given in Lemma 16. Let $\tau' \in \mathcal{SUB}(\mathrm{V}', \mathcal{T}'(\mathrm{V_{SIG}}'))$ and assume the condition of the implication of Lemma 16 to hold for this τ'. There exist $\tau \in \mathrm{INDSUB}(\mathrm{V}', cons')$ and $\sigma \in \mathcal{SUB}(\mathrm{V}'_{\mathrm{SIG}}, \mathcal{T}'(\mathrm{V_{SIG}}'))$ such that $\tau' = \tau\sigma$ and $\tau|_{\mathrm{V}} \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}(\mathrm{V_{SIG}}))$. By assumption we know that for each atom $(u = v)$ in Γ the formulas $(\mathrm{Def}\, u)$ and $(\mathrm{Def}\, v)$ are valid in $\mathcal{T}(\mathrm{V_{SIG}})/\leftrightarrow^*_{R,\mathrm{V_{SIG}}}$. Thus there exist $\hat{u}, \hat{v} \in \mathcal{GT}(cons)$ with $u\tau \leftrightarrow^*_{R,\mathrm{V_{SIG}}} \hat{u}$ and $v\tau \leftrightarrow^*_{R,\mathrm{V_{SIG}}} \hat{v}$. By Theorem 8 we obtain $u\tau \leftrightarrow^*_{R',\mathrm{V_{SIG}}'} \hat{u}$ and $v\tau \leftrightarrow^*_{R',\mathrm{V_{SIG}}'} \hat{v}$. Together with $u\tau' \leftrightarrow^*_{R',\mathrm{V_{SIG}}'} v\tau'$, which holds by assumption, this yields (exploiting stability of $\longrightarrow_{R,\mathrm{X}}$) $\hat{u} \leftrightarrow^*_{R',\mathrm{V_{SIG}}'} u\tau\sigma \leftrightarrow^*_{R',\mathrm{V_{SIG}}'} v\tau\sigma \leftrightarrow^*_{R',\mathrm{V_{SIG}}'} \hat{v}$. Thus (since $\leftrightarrow^*_{R,\mathrm{X}} \cap (\mathcal{GT}(cons)\times\mathcal{GT}(cons)) = \leftrightarrow^*_{R,\emptyset}$) we get $\hat{u} \leftrightarrow^*_{R',\emptyset} \hat{v}$ which, due to confluence of $\longrightarrow_{R',\emptyset}$, implies $\hat{u} \downarrow_{R',\emptyset} \hat{v}$. Theorem 8 yields $\hat{u} \downarrow_{R,\emptyset} \hat{v}$, hence (by monotonicity of $\longrightarrow_{R,\mathrm{X}}$ in X (Theorem 8)) $u\tau \leftrightarrow^*_{R,\mathrm{V_{SIG}}} \hat{u} \leftrightarrow^*_{R,\mathrm{V_{SIG}}} \hat{v} \leftrightarrow^*_{R,\mathrm{V_{SIG}}} v\tau$. Furthermore, for each atom $(\mathrm{Def}\, u)$ in Γ we have assumed that $(\mathrm{Def}\, u)$ is valid in $\mathcal{T}(\mathrm{V_{SIG}})/\leftrightarrow^*_{R,\mathrm{V_{SIG}}}$. Thus there exists some $\hat{u} \in \mathcal{GT}(cons)$ with $u\tau \leftrightarrow^*_{R,\mathrm{V_{SIG}}} \hat{u}$. Because of $R \models^D_{ind} \Gamma \longrightarrow \Delta$ we can conclude that there is an atom $(u = v) \in \Delta$ with $u\tau \leftrightarrow^*_{R,\mathrm{V_{SIG}}} v\tau$ or an atom $(\mathrm{Def}\, u) \in \Delta$ with $u\tau \leftrightarrow^*_{R,\mathrm{V_{SIG}}} \hat{u}$ for some $\hat{u} \in \mathcal{GT}(cons)$. Application of Theorem 8 yields $u\tau \leftrightarrow^*_{R',\mathrm{V_{SIG}}'} v\tau$ or $u\tau \leftrightarrow^*_{R',\mathrm{V_{SIG}}'} \hat{u}$, respectively. Finally, due to $\tau' = \tau\sigma$ (and stability of $\longrightarrow_{R,\mathrm{X}}$) we obtain $u\tau' \leftrightarrow^*_{R',\mathrm{V_{SIG}}'} v\tau'$ or $u\tau' \leftrightarrow^*_{R',\mathrm{V_{SIG}}'} \hat{u}$, respectively, as desired. Summarizing, we can conclude $R' \models^D_{ind} \Gamma \longrightarrow \Delta$ as was to be shown. ∎

Proof of (E): By Lemma 15(a), (D), Lemma 13(e). ∎

4 Discussion and Related Work

4.1 Advantages/Disadvantages

As shown above, all our notions of inductive validity have the desired monotonic behaviour w. r. t. consistent extension (under reasonable assumptions).

Concerning operational feasibility of the different types of inductive validity, the following can be said. Type-A can be approached by usual first-order theorem proving via induction schemes relying on the fact that only the validity of the inductive instances has to be considered. For type-B, C, and D we are and will be investigating inductive theorem proving techniques which are supported by a confluent rewriting relation. Lemma 16 allows us to develop powerful inference rules for type-D, some of which are not applicable for type-C (and B), which is no surprise since less formulas are of these types. Such an approach is more powerful than presenting an inference system for type-D only and then to approach type-C (and B) via Lemma 15(b) (and (c)). To see this, consider the specification with constructor constants a, b, c and non-constructor function symbol h with partial specification $h(a) = b,\ \ h(b) = a$. Now the formula $\mathrm{Def}\, h(c) \longrightarrow h(h(h(c))) = h(c)$ is type-C valid which cannot be inferred via Lemma 15(b). It is, however, easy to show its type-C validity via a type-C equivalence transformation into $x = h(c) \longrightarrow h(h(h(c))) = h(c)$ (for $x \in \mathrm{V}_{\mathrm{CONS}}$) and subsequent case analysis (via a "covering set" of substitutions for x) followed by "contextual rewriting". Note that the formula $h(h(h(c))) = h(c)$ is of type-D' only. In fact, there are examples for type-D' that require inferences of this kind which are far more complicated since they may not allow for a finite argumentation. Therefore, we suspect that it will be difficult to develop a prover that can effectively show those type-D' inductively valid formulas that do not result from type-C valid formulas $\Gamma \longrightarrow \Delta$ by deleting Def-atoms in Γ (cf. Lemma 13(c)).

4.2 Special Cases

In the sequel we will assume R to be a Def-MCTRS and $\rightarrow_{R,\mathrm{V}_{\mathrm{SIG}}}$ to be confluent.

Let us first consider the special case that $cons = sig$ and that no variables from $\mathrm{V}_{\mathrm{SIG}}$ occur in formulas. Note that, for $cons = sig$, our requirement of being constructor-based permits positive conditional equations only. In this case (disregarding formulas that involve Def-literals) type-A and B' validity are validity in all ($cons = sig$)-term-generated models. Furthermore, Type-B, C, D', D, and E coincide with validity in the unique minimal term-generated model (i. e. the initial model). Hence, we obtain classic initial semantics (for positive conditional equational specifications) as a special case of our general framework.

Still not permitting general variables in formulas, another important special case is that of sufficient completeness (cf. Lemma 13(g)), where type-C, D', D, and E again coincide, and where the model $\mathcal{GT}/\overset{*}{\leftrightarrow}_{R,\emptyset}$ (which establishes type-E) is $cons$-isomorphic to $\mathcal{GT}(cons)/\leftrightarrow^{*}_{R_{\mathbf{C}},\emptyset}$ (cf. Definition 4).

Let us finally restrict to positive formulas $\longrightarrow \Delta$. Here type-A and B' coincide. So do B, C, and D. Furthermore, these two groups coincide when no rule in R has a negative condition (cf. Lemma 15(d)).

4.3 Related Work

Let us now have a brief look at notions of inductive validity in the literature, most of which can be described as specializations within our framework. If we consider all symbols to be constructor symbols (and, consequently, forbid negative conditions in the rules), then we find type-A in [KR90] as well as in [BKR92]. The crucial idea of requiring the values of variables in formulas to be *defined*, i. e., to be substituted by constructor ground terms only, seems to appear first in [Zha88], [ZKK88]. The "constructor models" of [Zha88], [ZKK88], which are not models in the usual algebraic sense since functions are allowed to be partially interpreted, are consistently formalized in our framework by introducing the simple (order-sorted) notion of *sig*/*cons*-algebras. The notion of inductive validity of [ZKK88], [Zha88] can be described as type-A by implicitly interpreting all variables in rules as general variables and all variables in formulas as constructor variables. In [GS92] inductive validity is defined to be validity in the perfect model as introduced in [BG91]. This approach has a more general specification formalism permitting sets of first-order clauses instead of constructor-based PNCTRSs. The perfect model is determined as the least term-generated model w. r. t. some ground-total reduction ordering. The perfect model semantics, however, lacks the discussed monotonicity property. In [Pad90], [BL90], [KR88] and [BR93] we find the usual validity in the initial model which is like our type-E, assuming again all symbols to be constructor symbols and forbidding negative conditions in the rules. Kapur and Musser ([KM87], [KM86]) consider only unconditional equations and only congruences on ground terms (i. e. term-generated models) for validity, namely those that are maximally enlarged by random identification of undefined terms with defined ones (i. e. constructor ground terms) as long as this identification does not identify distinct constructor ground terms. Their intended congruence is then the intersection of all those maximally enlarged congruences. In [KM87] the maximal congruences are allowed to have some undefined terms left (accounting for the fact that the constructor-minimality requirement may forbid any further identification from some point on). The resulting "inductive models", however, lack the discussed monotonicity property, cf. [WGKP93] for an example. Therefore, in [KM86] the intersection is formed only over those congruences that have no undefined ground terms left. While we have no notion of inductive validity corresponding to that of [KM87], inductive validity in [KM86] coincides with type-D'. Finally, the problems due to not yet known or incompletely specified function symbols are also discussed in [Wal94], mainly along the lines of [KM86].

5 Conclusion

We have shown that considering inductive validity of first-order equational clauses instead of pure unconditional equations gives rise to various conceivable notions of inductive validity. Within the framework of constructor-based positive/negative conditional equational specifications (which provides an adequate unique model semantics) we have demonstrated that all these notions enjoy a desirable monotonic behaviour w. r. t. consistent extension, which is not the case for classic initial or perfect model validity.

Acknowledgements: We would like to thank Ulrich Kühler and Horst Prote for many valuable discussions and detailed criticisms on earlier versions of this paper, Jürgen Avenhaus and Klaus Becker for fruitful discussions, and Klaus Madlener for useful hints.

References

[AB92] J. Avenhaus, K. Becker. Conditional rewriting modulo a built-in algebra. SEKI-Report SR–92–11, FB Informatik, Univ. Kaiserslautern, 1992.

[Bac88] L. Bachmair. Proof by consistency in equational theories. In *Proc. 3rd IEEE Symposium on Logic in Computer Science*, pp. 228–233, 1988.

[Bec93] K. Becker. Semantics for positive/negative conditional rewrite systems. In M. Rusinowitch, J.L. Rémy, eds., *Proc. 3rd CTRS, LNCS 656*, pp. 213–225. Springer, 1993.

[BG91] L. Bachmair, H. Ganzinger. Perfect model semantics for logic programs with equality. In *Proc. 8th ICLP*, pp. 645–659. MIT Press, 1991.

[BKR92] A. Bouhoula, E. Kounalis, M. Rusinowitch. Automated mathematical induction. Rapport de Recherche 1663, INRIA, April 1992.

[BL90] E. Bevers, J. Levi. Proof by consistency in conditional equational theories. In S. Kaplan, M. Okada, eds., *Proc. 2nd CTRS, LNCS 516*, pp. 194–205. Springer, 1990.

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

[BR93] A. Bouhoula, M. Rusinowitch. Automatic case analysis in proof by induction. In *Proc. 13th IJCAI*, pp. 88–94, 1993.

[DJ90] N. Dershowitz, J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, ed., *Formal models and semantics, Handbook of Theoretical Computer Science*, vol. B, chapter 6, pp. 243–320. MIT Press, 1990.

[DOS88] N. Dershowitz, M. Okada, G. Sivakumar. Confluence of conditional rewrite systems. In S. Kaplan, J.-P. Jouannaud, eds., *Proc. 1st CTRS, LNCS 308*, pp. 31–44. Springer, 1988.

[GS92] H. Ganzinger, J. Stuber. Inductive theorem proving by consistency for first-order clauses. In M. Rusinowitch, J.-L. Rémy, eds., *Proc. of 3rd CTRS, LNCS 656*, pp. 226–241. Springer, 1993.

[Kap87] S. Kaplan. Positive/negative conditional rewriting. In S. Kaplan, J.-P. Jouannaud, eds., *Proc. 1st CTRS, LNCS 308*, pp. 129–143. Springer, 1987.

[KM86] D. Kapur, D. R. Musser. Inductive reasoning with incomplete specifications. Prelim. report. In *Proc. 1st LICS*, Cambridge, MA, 1986.

[KM87] D. Kapur, D. R. Musser. Proof by consistency. *Artificial Intelligence*, 31:125–157, 1987.

[KR88] E. Kounalis, M. Rusinowitch. On word problems in Horn theories. In E. Lusk, R. Overbeek, eds., *Proc. 9th CADE, LNCS 310* , pp. 527–535. Springer, 1988.

[KR90] E. Kounalis, M. Rusinowitch. Mechanizing inductive reasoning. In *Proc. 8th AAAI*, pp. 240–245. MIT Press, 1990.

[Pad90] P. Padawitz. Horn logic and rewriting for functional and logic program design. Technical Report MIP-9002, Univ. Passau, 1990.

[SNGM89] G. Smolka, W. Nutt, J.A. Goguen, J. Meseguer. Order-sorted equational computation. In H. Ait-Kaci, M. Nivat, eds., *Proc. CREAS*, vol. 2, Academic Press, 1989.

[Wal94] C. Walther. Mathematical induction. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, vol. 2, Clarendon Press, 1994.

[WG93] C.-P. Wirth, B. Gramlich. A constructor-based approach for positive/negative-conditional equational specifications. In M. Rusinowitch, J.-L. Rémy, eds., *Proc. of 3rd CTRS, LNCS 656*, pp. 198–212. Springer, 1993. Revised and extended version to appear in J. Symbolic Computation.

[WGKP93] C.-P. Wirth, B. Gramlich, U. Kühler, H. Prote. Constructor-based inductive validity in positive/negative-conditional equational specifications. SEKI-Report SR–93–05, FB Informatik, Univ. Kaiserslautern, 1993.

[Zha88] H. Zhang. *Reduction, Superposition and Induction: Automated Reasoning in an Equational Logic*. PhD thesis, Rensselaer Polytech. Inst., Troy, NY, 1988.

[ZKK88] H. Zhang, D. Kapur, M.S. Krishnamoorthy. A mechanizable induction principle for equational specifications. In E. Lusk, R. Overbeek, eds., *Proc. 9th CADE, LNCS 310*, pp. 162–181. Springer, 1988.