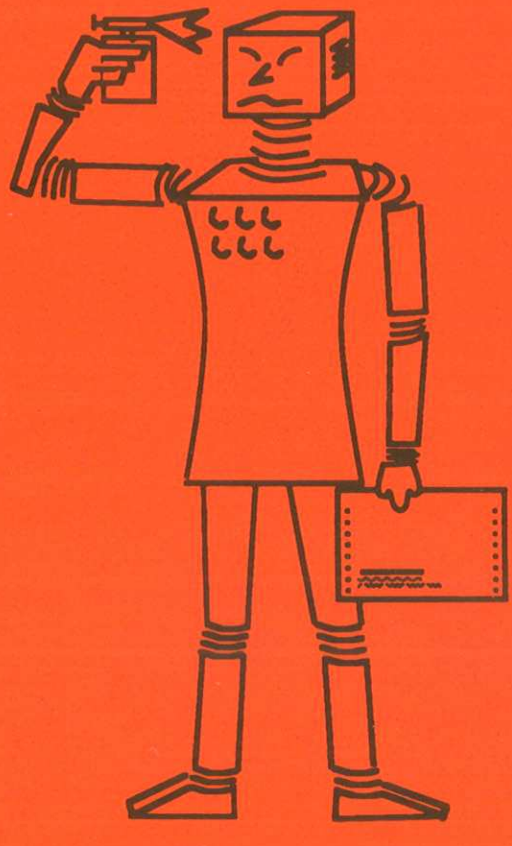
Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-6750 Kaiserslautern

SEKI - REPORT



Constructor-Based Inductive Validity in Positive/Negative-Conditional Equational Specifications

Claus-Peter Wirth, Bernhard Gramlich
Ulrich Kühler, Horst Prote

SEKI Report SR-93-05 (SFB)

Constructor-Based Inductive Validity in Positive/Negative-Conditional Equational Specifications

Claus-Peter Wirth, Bernhard Gramlich,
Ulrich Kühler, Horst Prote

Fachbereich Informatik, Universität Kaiserslautern,
D-67663 Kaiserslautern,
Germany

wirth@informatik.uni-kl.de

June 12, 1993

Abstract: We study algebraic specifications given by finite sets R of positive/negative-conditional equations (i. e. universally quantified first-order implications with a single equation in the succedent and a conjunction of positive and negative (i. e. negated) equations in the antecedent). The class of models of such a specification R does not contain in general a minimum model in the sense that it can be mapped to any other model by some homomorphism. We present a constructor-based approach for assigning appropriate semantics to such specifications. We introduce two syntactic restrictions: firstly, for a condition to be fulfilled we require the evaluation values of the terms of the negative equations to be in the constructor sub-universe which contains the set of evaluation values of all constructor ground terms; secondly, we restrict the constructor equations to have "Horn"-form and to be "constructor-preserving". A reduction relation for R is defined, which allows to generalize the fundamental results for positive-conditional rewrite systems, which does not need to be noetherian or restricted to ground terms, and which is monotonic w. r. t. consistent extension of the specification. Under the assumption of confluence, the factor algebra of the term algebra modulo the congruence of the reduction relation is a minimal model which is (beyond that) the minimum of all models that do not identify more objects of the constructor sub-universe than necessary and which establishes one of the four notions of inductive validity of Gentzen clauses we discuss. To achieve decidability of reducibility we define several kinds of compatibility of R with a reduction ordering and present a complete critical-pair test for the confluence of the reduction relation.

Keywords: Positive/negative-conditional specification, partial specification, constructor-based semantics, initial model, free model, inductive validity, constructor variables, order-sorted algebra, consistent extension, positive/negative-conditional rewriting, confluence, termination.

This research was supported by the Deutsche Forschungsgemeinschaft, SFB 314 (D4-Projekt)

Contents

1 Introduction and Overview

We present a constructor-based approach for assigning semantics to algebraic specifications with finite sets of positive/negative-conditional equations. In this approach, the non-constructor function symbols can be used for (possibly partially) defining functions on a domain of discourse supplied by the constructor terms and called the *constructor sub-universe*. For such partial definitions of functions, variables ranging over the constructor terms (or the constructor sub-universe) are likely to be more convenient than variables ranging over all terms (including "junk" terms) (or the whole universe), because the specifier usually[1] does not intend to tell how the functions behave on objects that are "*undefined*" in the sense that they do not belong to the domain of discourse. Therefore we generalize unconditional equations not only by adding positive and negative conditions but also by allowing *constructor variables* in addition to the usual *general variables.*

In general, specifications with positive/negative-conditional equations lack an initial model. The most promising attempt in literature to overcome this problem has been that in Kaplan[17]. There, one of the quasi-initial models is distinguished from the other models by means of control information extracted from the rules, which must be compatible with a noetherian ordering. In addition, Kaplan gives a straightforward ground term reduction relation. However, his approach violates the paradigm of separation of logic and control and does not allow to express the distinction of his quasi-initial model without the control part of the specification. For this reason (among others), we choose a new different approach. Instead of using control information we introduce two syntactic restrictions:

A. For a condition to be fulfilled, the terms of its negative equations must be "*defined*" in the sense that their evaluations fall into the constructor sub-universe. This requirement is achieved by adding condition literals expressing this property and goes well with our intention of taking the constructor sub-universe as the domain of discourse.

B. We restrict the constructor rules (which express equalities among the constructor terms) to have "Horn"-form and to be "constructor-preserving".

We then obtain the results of Kaplan[17] without using control information or noetherian orderings anymore. As a consequence, our reduction relation does not need to be noetherian or to be restricted to ground terms. Contrary to [17], we can show the monotonicity of this reduction relation w. r. t. consistent extension of the specification. As in [17], assuming confluence of our reduction relation, the factor algebra of the ground term algebra modulo the congruence of our reduction relation is a quasi-initial model for our specification. Unlike [17], however, it is also initial in the class of all models which do not identify more constructor ground terms than necessary.

To achieve decidability of reducibility and to enhance our means of testing for confluence, we define several kinds of compatibility of R with a reduction ordering, which enable us to present a complete critical-pair test for confluence of our reduction relation.

Finally, we define and disambiguate four notions of inductive validity of Gentzen clauses, compare them with notions found in literature, and show their monotonicity w. r. t. consistent extension of the specification.

The more difficult proofs of presented results can be found in appendix A.

[1] unless he wants to specify error-recovery or non-strict functions

2 Basic Notions and Notations

In this section we describe our non-standard basic notions and notations for sets and terms. More basic notions and notations (for substitutions, algebras, orderings, and reduction relations) can be found in sect. 4.

2.1 Sets and Classes, Multi-sets

We use the following non-standard notation for classes A, B:

$$\begin{array}{rcll}
\mathbb{N}_+ & := & \{\, n \in \mathbb{N} \mid n \neq 0 \,\} & \\
\mathrm{dom}(A) & := & \{\, a \mid \exists b : (a,b) \in A \,\} & (\textit{domain}) \\
\mathrm{ran}(A) & := & \{\, b \mid \exists a : (a,b) \in A \,\} & (\textit{range}) \\
\mathrm{field}(A) & := & \mathrm{dom}(A) \cup \mathrm{ran}(A) & \\
A(B) & := & \bigcap_{(B,C)\in A} C & (\textit{function application of } A)^{2} \\
A[B] & := & \{\, b \mid \exists a \in B : (a,b) \in A \,\} & (\textit{relation application of } A) \\
\mathcal{F}(A) & := & \{\, S \mid S \subseteq A \ \wedge \ (S \text{ is finite}) \,\} & \\
A^{*} & := & \{\, w : \{0,1,\ldots,n-1\} \to A \mid n \in \mathbb{N} \,\} & (\textit{set of words}) \\
A^{+} & := & \{\, w : \{0,1,\ldots,n-1\} \to A \mid n \in \mathbb{N}_+ \,\} & (\textit{set of nonempty words}) \\
\mathrm{id} & := & \{\, (a,a) \mid a \text{ is a set} \,\} & (\textit{identity-function}) \\
\mathcal{U} & := & \mathrm{dom}(\mathrm{id}) & (\textit{class of all sets}) \\
A^{-1} & := & \{\, (b,a) \mid (a,b) \in A \,\} & (\textit{reverse of a relation}) \\
A \circ B & := & \{\, (a,c) \mid \exists b : \ ((a,b) \in A \ \wedge \ (b,c) \in B) \,\} & (\textit{product of relations})^{3} \\
A^{\circ(0)} & := & \mathrm{id} & \\
A^{\circ(i+1)} & := & A^{\circ(i)} \circ A \quad \text{for } i \in \mathbb{N} & (\textit{power of a relation}) \\
A^{\circledast} & := & \bigcup_{i\in\mathbb{N}} A^{\circ(i)} & \left(\begin{array}{c}\mathcal{U}\textit{-reflexive, transitive} \\ \textit{closure of a relation}\end{array}\right) \\
A^{\oplus} & := & \bigcup_{i\in\mathbb{N}_+} A^{\circ(i)} & \left(\begin{array}{c}\textit{transitive} \\ \textit{closure of a relation}\end{array}\right)
\end{array}$$

We use $\emptyset$ both for the empty set and the empty word. For application of the function f to i we write f_i besides $f(i)$, such that w_i denotes the[4] i^{th} "letter" of a word w. Speaking of a *relation* we always think of a binary relation unless indicated otherwise. We use $\uplus$ for the union of disjoint sets.

Multi-sets differ from sets only in their possibility to contain an element more than once. We denote them very similar to sets: $\emptyset$ denotes the empty multi-set as well as the empty set. We use '$\langle$' instead of '$\{$', '$\rangle$' instead of '$\}$', '$\sqsubseteq$' instead[5] of '$\subseteq$', and '$\sqcup$' instead[6] of '$\cup$'.

---

[2] As this set-theoretic definition might be difficult to understand, here is an equivalent one:

$$A(B) = \begin{cases} C & \text{if } \ B \in \mathrm{dom}(A) \ \wedge \ \forall D : ((B,D) \in A \ \Rightarrow \ D = C) \\ \mathcal{U} & \text{if } \ B \notin \mathrm{dom}(A) \\ \textit{“rubbish”} & \text{otherwise} \end{cases}$$

[3] This is very convenient for relations because $a\,A\,b\,B\,c$ implies $a\,(A \circ B)\,c$. It is, however, not generally considered the most convenient definition when A and B are used as functions because $B(A(a)) = b$ (and not $A(B(a)) = b$) implies $(A \circ B)(a) = b$, provided that $a \in \mathrm{dom}(A)$ and $A(a) \in \mathrm{dom}(B)$.

[4] Notice that we start with 0 instead of 1

[5] $A \sqsubseteq B$:iff each element occurring n-times in A occurs at least n-times in B

[6] Each element occurring exactly n-times in A and exactly m-times in B occurs exactly $(n+m)$-times in $A \sqcup B$.

$\langle\, f(i) \mid i \in I \,\rangle$ denotes the multi-set containing an element 'a' exactly $|\{\, i \in I \mid f(i) = a \,\}|$-times. FMul($S$) denotes the set of finite multi-sets whose elements are in the set S.

It would be nice to keep the multi-sets as abstract as they are now. But as we don't want to introduce a duplicate of set-theoretic mathematics for multi-sets we have to be more concrete: We assume a multi-set to be a set that contains for each multi-set-element a occurring n-times in it n different representation set-elements[7] $\hat{a}_0, \ldots, \hat{a}_{n-1}$ with $\forall i < n : a = \mathrm{set}(\hat{a}_i)$. To avoid confusion we don't introduce an analogue of '$\ldots \in \ldots$' for multi-sets, but write '$\ldots \in \mathrm{set}[\ldots]$' instead.

2.2 Terms

We will consider terms of fixed arity over many-sorted signatures. A *signature*

$$\mathrm{sig} = (\mathsf{F}, \mathrm{S}, \alpha)$$

consists of an enumerable set of function symbols F, a finite set of sorts S (disjoint from F), and a computable arity-function $\alpha : \mathsf{F} \to \mathrm{S}^{+}$. For $f \in \mathsf{F}$: $\alpha(f)$ is the list of argument sorts augmented by the sort of the result of f; to ease reading we will add a ' $\longrightarrow$ ' after a nonempty list of argument sorts. A *constructor sub-signature of the signature* sig is a signature

$$\mathrm{cons} = (\mathsf{C}, \mathrm{S}, \alpha|_{\mathsf{C}})$$

such that the set C is a decidable subset of F. C is called the set of *constructor symbols*; the complement $\mathsf{N} = \mathsf{F} \setminus \mathsf{C}$ is called the set of *non-constructor symbols.*

Example 2.1 (Signature with Constructor Sub-signature)

C	=	{0, s, false, true, nil, cons}		
N	=	{−, memberp, }		
S	=	{nat, bool, list}		
α(0)	=			nat
α(s)	=	nat	⟶	nat
α(false)	=			bool
α(true)	=			bool
α(nil)	=			list
α(cons)	=	nat list	⟶	list
α(−)	=	nat nat	⟶	nat
α(memberp)	=	nat list	⟶	bool

A *variable-system for a signature* sig is an S-sorted family of decidable sets of variable symbols which are mutually disjoint and disjoint from F. As the basis for our terms throughout the whole paper we assume two fixed disjoint variable-systems $\mathrm{V}_{\mathrm{SIG}}$ of *general variables* and $\mathrm{V}_{\mathrm{CONS}}$ of *constructor variables* such that for each $s \in \mathrm{S}$ we have $|\mathrm{V}_{\mathrm{SIG},s}|, |\mathrm{V}_{\mathrm{CONS},s}| \notin \mathbb{N}$. By abuse of notation we will use the symbol 'X' for an S-sorted family not only to denote the family $X = (X_s)_{s \in \mathrm{S}}$ itself, but also the union of its *ranges*: $\bigcup_{s \in \mathrm{S}} X_s$. $\mathcal{T}(\mathrm{sig}, \mathrm{V}_{\mathrm{SIG}} \uplus \mathrm{V}_{\mathrm{CONS}})$

[7] generated in some fixed manner (e. g. $\hat{a}_i := (a, i)$) for intensional equality of multi-sets to imply extensional equality.

denotes the S-sorted family of all well-sorted *(variable-mixed) terms* over sig/$\mathrm{V_{SIG}} \uplus \mathrm{V_{CONS}}$, while $\mathcal{GT}(\mathrm{sig})$ denotes the S-sorted family of all well-sorted *ground terms* over sig. Similarly, $\mathcal{T}(\mathrm{cons}, \mathrm{V_{SIG}} \uplus \mathrm{V_{CONS}})$ denotes the S-sorted family of all *(variable-mixed) constructor terms*, $\mathcal{T}(\mathrm{cons}, \mathrm{V_{CONS}})$ denotes the S-sorted family of all *pure constructor terms*, while $\mathcal{GT}(\mathrm{cons})$ denotes the S-sorted family of all *constructor ground terms*. For abstraction from problems with empty sorts, we always assume $\mathcal{GT}(\mathrm{cons})$ to have nonempty ranges only. We will denote all terms in prefix-form without any confusing notational sugar (such as parentheses or commas).

As exhibited in Avenhaus&Becker[1], our terms can be looked upon (and we will do so) as family of order-sorted terms in the style of Gogolla[15] or Smolka&al.[25]: Take $\{\mathrm{SIG}, \mathrm{CONS}\} \times \mathrm{S}$ for the sorts with the sort declaration that for each $s \in \mathrm{S}$ the sort (CONS, s) is a sub-sort of the sort (SIG, s); and replace each arity declaration of the form $\alpha(f) = s_0 \ldots s_{n-1} \longrightarrow s_n$ by the arity declaration

$$\alpha(f) \ni (\mathrm{SIG}, s_0) \ldots (\mathrm{SIG}, s_{n-1}) \longrightarrow (\mathrm{SIG}, s_n);$$

moreover, for $f \in \mathrm{C}$ add the arity declaration

$$\alpha(f) \ni (\mathrm{CONS}, s_0) \ldots (\mathrm{CONS}, s_{n-1}) \longrightarrow (\mathrm{CONS}, s_n).$$

A *variable-system for signature* sig *with sub-signature* cons is defined to be a $\{\mathrm{SIG}, \mathrm{CONS}\} \times \mathrm{S}$-sorted family $\mathrm{V} = (\mathrm{V}_{\varsigma,s})_{(\varsigma,s) \in \{\mathrm{SIG},\mathrm{CONS}\} \times \mathrm{S}}$ of decidable sets which are mutually disjoint and disjoint from F. For denoting the $\{\mathrm{SIG}, \mathrm{CONS}\} \times \mathrm{S}$-sorted family of variables occurring in a structure A (which might be a term or a list of terms, e. g.) we will use $\mathcal{V}(A)$. For denoting the S-sorted family of general (resp. constructor) variables occurring in A we will use $\mathcal{V}_{\mathrm{SIG}}(A)$ (resp. $\mathcal{V}_{\mathrm{CONS}}(A)$).

Now the order-sorted notation for our sets of terms are the $\{\mathrm{SIG}, \mathrm{CONS}\} \times \mathrm{S}$-sorted families $\mathcal{T} = (\mathcal{T}_{\varsigma,s})_{(\varsigma,s) \in \{\mathrm{SIG},\mathrm{CONS}\} \times \mathrm{S}}$ and $\mathcal{GT} = (\mathcal{GT}_{\varsigma,s})_{(\varsigma,s) \in \{\mathrm{SIG},\mathrm{CONS}\} \times \mathrm{S}}$ given by $\mathcal{T}_{\mathrm{SIG},s} := \mathcal{T}(\mathrm{sig}, \mathrm{V_{SIG}} \uplus \mathrm{V_{CONS}})_s$, $\mathcal{T}_{\mathrm{CONS},s} := \mathcal{T}(\mathrm{cons}, \mathrm{V_{CONS}})_s$, $\mathcal{GT}_{\mathrm{SIG},s} := \mathcal{GT}(\mathrm{sig})_s$, and $\mathcal{GT}_{\mathrm{CONS},s} := \mathcal{GT}(\mathrm{cons})_s$ for each $s \in \mathrm{S}$. To avoid confusion: Note that $\mathcal{T}_{\mathrm{CONS},s} \subseteq \mathcal{T}_{\mathrm{SIG},s}$ for $s \in \mathrm{S}$, whereas $\mathrm{V}_{\mathrm{CONS},s} \cap \mathrm{V}_{\mathrm{SIG},s} = \emptyset$. Our custom of reusing the symbol of a family for the union of its ranges now allows to write $\mathcal{T}$ as a shorthand for $\mathcal{T}(\mathrm{sig}, \mathrm{V_{SIG}} \uplus \mathrm{V_{CONS}})$.

All in all, this leads to a simple treatment (also of subsequent notions) of our approach within the order-sorted framework. For the sake of simplicity and flexibility, however, we have also presented a more direct approach here.

An *occurrence* (or *position*) is an element of $\mathbb{N}_+^*$. We write $p|q$ to express that neither p is a prefix of q, nor q a prefix of p. For a term $t \in \mathcal{T}$ we define the *set of its occurrences* (or *positions*) $\mathcal{O}(t)$ recursively by: $\mathcal{O}(x) = \{\emptyset\}$ $(x \in \mathrm{V})$ and

$$\mathcal{O}(ft_1 \ldots t_n) = \{\emptyset\} \cup \bigcup_{1 \le i \le n} \{\, iq \mid q \in \mathcal{O}(t_i) \,\} \quad (f \in \mathrm{F};\ t_1, \ldots, t_n \in \mathcal{T}).$$

For a term t we denote by t/p the subterm of t at occurrence p and by $t[p \leftarrow t']$ the result of replacing t/p with t' at occurrence p in t. For $P \subseteq \mathcal{O}(t)$, $\forall p,q \in P : (p \neq q \Rightarrow p|q)$, we denote by $t[p \leftarrow t'_p \mid p \in P]$ the result of replacing for each $p \in P$ the subterm at occurrence p in the term t with the term t'_p. A term $t \in \mathcal{T}$ is *linear* :iff $\forall p,q \in \mathcal{O}(t) : (t/p = t/q \in \mathrm{V} \Rightarrow p = q)$; it is *reducible* w. r. t. a relation $\Longrightarrow$:iff $t \in \mathrm{dom}(\Longrightarrow)$.

We define the following sets of equations:

$\mathrm{Eq}(\mathrm{sig}, \mathrm{X}) := \{\, \langle t, t' \rangle \mid \exists s \in \mathrm{S} : t, t' \in \mathcal{T}(\mathrm{sig}, \mathrm{X})_s \,\}$ *"set of undirected equations"*

$\mathrm{DEq}(\mathrm{sig}, \mathrm{X}) := \{\, (t, t') \mid \exists s \in \mathrm{S} : t, t' \in \mathcal{T}(\mathrm{sig}, \mathrm{X})_s \,\}$ *"set of directed equations"*

3 Motivation

Just as with unconditional equations, the class of algebras satisfying a set R of positive-conditional equations[8] contains an initial algebra, whereas reduction is more difficult because the equations only have to hold if their condition holds[9]: The replacement of the left-hand side of an instantiated rule by its right-hand side is not known to be correct until the condition of the instantiated rule has been shown to hold in the initial model of R. Further problems occur when one allows negated equations among the equations of the conjunction that forms the condition. Kaplan[17] defines a [negated] equation in the condition to hold :iff its terms [do not] have a common reduct (w. r. t. $\stackrel{\circledast}{\Longrightarrow}$). If the resulting reduction relation is confluent and the rules are *decreasing* (cf. [12]) w. r. t. some reduction ordering $>$, then its congruence closure is minimal (but not a minimum!) w. r. t. set-inclusion among the congruence relations whose factor algebra (w. r. t. the ground term algebra $\mathcal{GT}$) is a model[10] of R. Despite of the lack of an initial model even in this restricted case, positive/negative-conditional equations are really necessary for convenient specification, as can be seen by the following example, where '$\longleftarrow$' precedes the condition of an equation.

Example 3.1 (continuing Example 2.1)

R:	$-\,x\,0$	$=$	x		
	$-\,\mathsf{s}\,x\,\mathsf{s}\,y$	$=$	$-\,x\,y$		
	memberp x nil	$=$	false		
	memberp x cons y l	$=$	true	$\longleftarrow$	$x = y$
	memberp x cons y l	$=$	memberp x l	$\longleftarrow$	$x \neq y$

The Importance of Confluence

First we are going to explain why confluence is essential for reduction with positive/negative-conditional rules: Firstly (even without negative equations), confluence is needed for the completeness of testing semantic equality of two condition terms by looking for a common reduct[11]. This means: We need confluence for the congruence defined in Kaplan[17] to yield a model of R. Secondly, it is needed for guaranteeing the congruence to be minimal, as can be seen from:

Example 3.2
Let a, b, c, d, e be constants of the same sort, $a > b > c > d > e$.

R: $c = d$
$c = e$
$a = b \;\longleftarrow\; e \neq d$

Then, for the reduction relation of Kaplan[17]:

$\Longrightarrow\; = \;\{\ (c,d),\ (c,e),\ (a,b)\ \}$, *whose congruence closure* $\stackrel{\circledast}{\Longleftrightarrow}$ *is not minimal.*

---

[8] By a positive-conditional equation we mean an equation that has associated with it a condition that consisting of a (possibly empty) conjunction of positive (i. e. not negated) equations. For a detailed treatment of positive-conditional equations cf. Kaplan[16].

[9] i. e. $\forall \vec{x} : (\text{equation}(\vec{x}) \Leftarrow \text{condition}(\vec{x}))$

[10] taking the conditional equations as universally quantified formulas of first order equational logic (as indicated in footnote 9) and using standard algebra model semantics

[11] w. r. t. the reflexive and transitive closure $\stackrel{\circledast}{\Longrightarrow}$ of $\Longrightarrow$

While confluence can be dropped for merely positive conditional equations by testing for congruence instead of testing for the existence of a common reduct of two condition terms, it is worse for positive/negative-conditional equations: It does not suffice to test non-congruence for inequality of two condition terms if confluence is not provided, as can be seen from:

Example 3.3
Let the signature and the ordering be as in the previous example.

$$\begin{aligned} \text{R:}\quad & c = d \longleftarrow d \neq e \\ & c = e \end{aligned}$$

Any congruence yielding a model of R must contain (d, e): If it did not, it would contain (c, d) and (c, e), and hence (d, e). Therefore, no matter which congruence we actually use for condition-testing, the test of $d{\neq}e$ with such a (model-yielding) congruence will always fail, such that we cannot establish $c \overset{\circledast}{\Longleftrightarrow} d$ by testing the condition of the first rule. But R has the minimum model '$c{=}d{=}e$', which cannot be obtained by the simple method of condition-testing anymore, but only by paramodulation and factoring instead, which in our opinion are too complicated for establishing just a simple reduction step.

By this we conclude that in case of negative equations in the condition, computation of a correct reduct by the method of condition-testing is only possible if confluence is provided.

Criticism of Kaplan[17]

The major shortcoming of the reduction relation in [17], however, is (as noted above) that its congruence closure is not a minimum[12] but only minimal[13] among the congruences yielding a model of R. There might be reductions $s \Longrightarrow t$ with $s{=}t$ not holding in all models logically specified by R. Kaplan[17] argues as follows:

By writing ' $c{=}d \longleftarrow d{\neq}e$ ' instead of the logically equivalent ' $c{=}d \vee d{=}e$ ' the specifier adds some "operational" information to control the choice of the intended minimal congruence '$c{=}d$' of the congruences yielding a model of R ('$c{=}d$', '$d{=}e$', and '$c{=}d{=}e$').

But writing ' $c{=}d \vee d{=}e$ ' in the form of ' $c{=}d \longleftarrow d{\neq}e$ ' is actually motivated by the ordering aspect and not by the specifier's[14] intention which of all minimal models to choose. What's worse is that semantics is given by control and not expressible without; thereby violating the paradigm of separation of logic and control: R is not a *logical* specification (suitable for computation) anymore (as it was the case with positive-conditional equations), but a program with none but operational semantics. Therefore we lose the nice aspects of the pure logic view. To name one: The monotonicity of logic is lost: If we, e. g., complete the specification of a partially specified function[15] in a proper way[16], we might destroy some reductions and congruences[17] that were possible before:

[12] i. e. being smaller than anything else

[13] i. e. there is nothing smaller

[14] Who is very likely to be unable to keep track of all consequences of his pieces of "operational" information

[15] And partially specified functions are very useful in expressing exactly what is required by the specifier!

[16] i. e. not confusing different constructor terms

[17] by this we mean elements of $\overset{\circledast}{\Longleftrightarrow}$

Example 3.4 (continuing Example 3.1)
memberp 0 cons − 0 s 0 nil $\stackrel{\circledast}{\Longrightarrow}$ false *no longer holds after adding the rule* $- 0\, \mathsf{s}\, x = 0$.

Similarly, reduction of non-ground terms is of no use because the reduction relation is not stable:

Example 3.5 *As* x *does not reduce to* 0 *, one might say*

$$\mathsf{memberp}\ 0\ \mathsf{cons}\ x\ \mathsf{nil} \stackrel{\circledast}{\Longrightarrow} \mathsf{false}\ .$$

But for $x \mapsto 0$ *this does not make any sense.*

The *perfect model* semantics approach of Bachmair&Ganzinger[4], which also includes a completion procedure, generalizes Kaplan's approach [17] by abstracting the control information hidden in the syntactic form of rules into a reduction ordering which must be total on ground terms and which determines the construction process of perfect models. With our approach, however, reduction orderings are not at all needed for defining semantics. Cf. Becker[6] for the exact interrelation between the three approaches of Kaplan[17], Bachmair&Ganzinger[4], and us.

Looking for Remedy

One could think that the problem of a minimal congruence not being a minimum hardly arises or is avoidable by convenient purely syntactic restrictions on the defining rules. The following example will exhibit that such restrictions for the congruence of [17] cannot be reasonable in practice of specification:

Example 3.6 (continuing Example 3.1)
We exclude[18] *the function symbols* 0, s, *and* − *(together with their respective rules), and enrich the signature with the following two constants:* $\alpha(\mathsf{a}) = \alpha(\mathsf{b}) = \mathsf{nat}$ *. The reduction relation* $\Longrightarrow$ *of [17] is confluent*[19] *and noetherian*[20] *in this case.*

Consider the following two congruence relations on ground terms, given by their congruence classes:

$\stackrel{\circledast}{\Longleftrightarrow}$: {a}
{b}
{false} ∪ {memberp x l |($x \in$ {a, b} ∧ (x *doesn't occur in* l))}
{true} ∪ {memberp x l |($x \in$ {a, b} ∧ (x *does occur in* l))}
{nil}
{cons a nil}
{cons b nil}
{cons a cons a nil}
{cons a cons b nil}
⋮

[18]It's not that they make trouble. We just omit them to make the example simpler.
[19]There are no feasible critical pairs
[20]Use the lexicographic path ordering given by memberp ≻ true , false

$\underline{\sim:}$ { a, b }
{ false , memberp a nil , memberp b nil }
{ true } ∪ { memberp x l $|(x \in \{$a, b$\}$ ∧ $l \neq$ nil)}
{ nil }
{ cons a nil , cons b nil }
{ cons x cons y nil | $x, y \in \{$a, b$\}$ }
⋮

Now, both $\overset{\circledast}{\Longleftrightarrow}$ *and* $\sim$ *yield a model of* R. *By* $a \sim b$; $a \overset{\circledast}{\not\Longleftrightarrow} b$; *we know that* $\sim$ *is not a minimum. By* memberp a cons b nil $\overset{\circledast}{\Longleftrightarrow}$ false *and* memberp a cons b nil $\not\sim$ false *we know that* $\overset{\circledast}{\Longleftrightarrow}$ *isn't a minimum either. But both* $\overset{\circledast}{\Longleftrightarrow}$ *and* $\sim$ *are minimal among the congruences that yield a model of* R. *Hence their intersection does not yield a model of* R.

Notice that the example is really a standard specification example and not a sophisticated one. Therefore, restrictions on the structure of the rules for avoiding minimal non-minimum models would necessarily forbid such basic and essential specifications as the memberp-example above.

Thus, we have to choose between a $\neq$ b and memberp a cons b nil $\neq$ false . As $\overset{\circledast}{\Longleftrightarrow}$ is somehow more appealing than $\sim$, one may argue that a $\neq$ b is somewhat more important than memberp a cons b nil $\neq$ false by stating a, b to be constructors and thinking freeness of constructors to be more important than that of non-constructors. But this treatment does not solve the problem in general: If

1. a or b is changed into a (composite) non-constructor term,

or

2. memberp is stated to be a constructor symbol too,

then the very same problem arises again.

Our Solution

Now, while the simple attempt above fails, the intended bias towards freeness of constructor terms can be achieved with the help of a new unary predicate 'Def' (cf. sect. 4) (which is also necessary for sufficient expressibility of lemmas for inductive theorem proving[21]) in the fol. way:

A. Adding condition literals expressing *definedness* for all terms of negative equations in the condition. A term t is *defined* <u>:iff</u> 'Def t' holds <u>:iff</u> t has a congruent constructor ground term. For our example above this means that the last memberp-rule is not applicable if a or b is undefined, thereby avoiding the problem of (1) above.

B. Forcing rules whose left-hand sides are constructor terms to have no negative equations in their conditions and to be constructor-preserving[22]. For our example above, this means that 'memberp' cannot be a constructor symbol, thereby avoiding the problem of (2) above.

---

[21] Cf. Wirth[28]. Lemmas of the form " Def f $x_0 \ldots x_{n-1}$ " (with the x_i being different constructor variables), expressing that the symbol 'f' denotes a "*total*" function, are very important for inductive theorem proving.

[22] i. e.: all terms in such a "constructor rule" are (variable-mixed) constructor terms and all variables of a constructor rule occur in its left-hand side

(B) is purely syntactic and not very restrictive in practice as it only limits congruences between constructor terms (and this even less restrictively than usual). (A) is not a usage of control information as before. It just means that '$\neq$' is syntactically restricted to defined terms.

Undefined terms are due to incomplete knowledge about the model world or partially specified functions. In this context, functions are *partial*[23] not because the specifier has explicitly stated their partiality as a property of importance, but because he has partially left open their definition, maybe due to partial information, due to irrelevance of the functions' further behaviour for the specification in the current state of development, or even due to partiality being actually intended. Thus, partiality and undefinedness are not part of the specification but a result from its incompleteness. For this reason, the undefined terms are often thought to be equal to some unknown constructor ground terms:

Kapur&Musser[18, 19] consider those congruences which are maximally enlarged by random identification of undefined terms with constructor ground terms, as long as this identification does not identify two distinct constructor ground terms. Their intended congruence is then the intersection of all those maximally enlarged congruences. In [18] the maximal congruences are allowed to have some undefined terms left; this causes the problem that one cannot describe the intended congruence by model semantics[24]. Therefore in [19] the intersection is done only over those congruences that have no undefined terms left: These congruences can easily be described in terms of model semantics: A model $\mathcal{A}$ is additionally[25] required to satisfy the following: Let $\overset{\circledast}{\Longleftrightarrow}$ denote the initial congruence[26] of R and k its canonical[27] cons-epimorphism from $\mathcal{GT}(\mathrm{cons})$ to $\mathcal{GT}(\mathrm{cons})/\overset{\circledast}{\Longleftrightarrow}$. Now the unique cons-homomorphism[28] h from $\mathcal{GT}(\mathrm{cons})/\overset{\circledast}{\Longleftrightarrow}$ to $\mathcal{A}$ given by $kh = (\mathcal{A}|_{\mathcal{GT}(\mathrm{cons})_s})_{s\in S}$ is required to be an isomorphism. A third way of removing the undefined terms is to require h to be epimorphic instead of isomorphic, i. e. $\mathcal{A}$ is additionally[25] required to be cons-term-generated. While the theory of the last two attempts is very beautiful, the resulting congruences may be very difficult to understand: One needs a very sophisticated way of argumentation for showing two terms equal — even for some very simple examples.

Based on this tradition of thinking undefined terms to be possibly equal to constructor ground terms, the above item (A) of our approach can be justified the fol. way:

---

[23] by which we mean a function with symbol say 'f', for which the application 'f t_0 ... t_{n-1}' to some constructor ground terms $t_0, \ldots, t_{n-1}$ is an undefined term.

[24] Of course, this is tried to be done in [18]. But their "inductive model" (which is defined to be a model with free constructors whose proper epimorphic images are no models with free constructors) is a nasty thing: Normally, a model uses to keep being a model when one throws away some equations of the specification, thereby establishing the monotonicity of logic. The "inductive models" do not have this property. To see this take $\mathsf{C} = \{\mathsf{false}, \mathsf{true}, 0\}$; $\mathsf{N} = \{\mathsf{s}, \mathsf{zerop}\}$; $\mathrm{R} = \{\ \mathsf{zerop}\,0 = \mathsf{true},\quad \mathsf{zerop}\,\mathsf{s}\,x = \mathsf{false}\ \}$. Now the following $\mathcal{A}$ is an "inductive model" for R but not for $\emptyset$ (where we need $|\mathcal{A}(\mathsf{nat})| = 1$): $\mathcal{A}(\mathsf{bool}) = \{\mathsf{FALSE}, \mathsf{TRUE}\}$; $\mathcal{A}(\mathsf{nat}) = \{0, 1\}$; $\mathsf{true}^{\mathcal{A}} = \mathsf{TRUE}$; $\mathsf{false}^{\mathcal{A}} = \mathsf{FALSE}$; $0^{\mathcal{A}} = 0$; $\mathsf{s}^{\mathcal{A}}(x) = 1$; $\mathsf{zerop}^{\mathcal{A}}(0) = \mathsf{TRUE}$; $\mathsf{zerop}^{\mathcal{A}}(1) = \mathsf{FALSE}$.
We can also see by this that we indeed have no monotonic logic here: $\emptyset \vdash 0 = \mathsf{s}\,0$; but (as seen by $\mathcal{A}$): $\mathrm{R} \nvdash 0 = \mathsf{s}\,0$.

[25] besides making true the universally quantified equations of R

[26] which exists because [18, 19] consider unconditional equations only

[27] sometimes called ***natural*** instead of canonical; mapping each term of $\mathcal{GT}(\mathrm{cons})$ to its congruence class.

[28] given by the Homomorphism-Theorem(4.2)

- *Considering dynamic extension of specifications:* If two terms can be shown equal by $\overset{\circledast}{\Longleftrightarrow}$, they will keep being equal even if an undefined term will be identified with a defined term later on.[29] On the other hand might an undefined term become equal (w. r. t. $\overset{\circledast}{\Longleftrightarrow}$) to a previously unequal term when identifying an undefined term with a defined term. Thus, we had better be cautious: We should not pretend to be able to distinguish something undefined from anything else (as the former might in the sequel be defined to be the latter).

- *From a static point of view on the specification:* Two distinct terms may be equal or unequal, no matter whether they are defined or undefined. In particular may an undefined term be both unequal to some distinct undefined term and equal to some other. This inequality between undefined terms, however, differs from the inequality between defined terms in that it is not considered sufficient for the fulfilledness of an inequality literal in the condition of an equation. This means that we have a "*closed world assumption*" which is restricted to the constructor ground terms, saying that two constructor ground terms are meant to be unequal unless their equality is specified by the constructor rules. According to this, we use "*negation as failure*" on the defined terms only, and not on the undefined terms where the specification is allowed to be incomplete and open.

In the fol. sections we will show that by the little changes of (A), (B), we get a straightforward reduction relation $\Longrightarrow$ that has the following advantages (compared to the one of Kaplan[17]) (cf. sect. 6):

1. Its congruence closure $\overset{\circledast}{\Longleftrightarrow}$ yields a model that is not only minimal but also the (up to isomorphism) uniquely determined minimum among those sig-term-generated models of R that do not identify more constructor ground terms than necessary (provided (as also required for $\overset{\circledast}{\Longleftrightarrow}$ of [17] being minimal) that $\Longrightarrow$ is confluent).

2. $\Longrightarrow$ is monotonic w. r. t. the addition of new rules that do not have left-hand sides which are old constructor terms.

3. $\Longrightarrow$ is stable when defined also on non-ground terms.

As shown in the examples above, the reduction relation of [17] has none of these properties. We will now revisit these examples to illustrate how our restrictions solve their problems.

1. (Example 3.6). If a and b are defined terms, then $\overset{\circledast}{\Longleftrightarrow}$ becomes the *minimum* among those congruences which do not identify more constructor ground terms than necessary. Contrariwise, if a or b is undefined, then the intersection of $\overset{\circledast}{\Longleftrightarrow}$ and $\sim$ becomes a model of R because the last memberp-rule now reads
$$\mathsf{memberp}\ x\ \mathsf{cons}\ y\ l = \mathsf{memberp}\ x\ l \longleftarrow x \neq y\ ,\ \mathrm{Def}\ x\ ,\ \mathrm{Def}\ y\ ;$$
thus, memberp a cons b nil is neither true nor false now, but undefined instead.

2. (Example 3.4). We do not have $\mathsf{memberp}\ 0\ \mathsf{cons} - 0\ \mathsf{s}\ 0\ \mathsf{nil} \overset{\circledast}{\Longrightarrow} \mathsf{false}$ anymore: memberp 0 cons − 0 s 0 nil is irreducible because − 0 s 0 is undefined.

3. (Example 3.5). As x is undefined: $\mathsf{memberp}\ 0\ \mathsf{cons}\ x\ \mathsf{nil} \overset{\circledast}{\not\Longrightarrow} \mathsf{false}$.

[29] Cf. Theorem 6.16

Furthermore, if our reduction relation is confluent and there aren't any undefined ground terms[30], then there is no difference between our ground reduction relation and that of Kaplan[17]. Therefore in this important case we offer semantics for the reduction relation of [17] not using any control information.

Finally, we are not only able to specify our semantics without using control information, but also able to remove the control aspect of requiring (for admissibility of a specification) the rules to be decreasing. As a consequence, our reduction relation does not need to be noetherian or restricted to ground terms.

Two types of variables

An additional feature of our presentation here, is our distinction between two kinds of variables. While the distinction between constructor terms and general terms is commonly accepted and considered fruitful, our distinction between constructor variables and general variables may require some explanation: General variables may be substituted by any term of the whole signature. Constructor variables, however, may only be substituted by pure constructor terms consisting of constructor function and constructor variable symbols. In the field of model semantics, this distinction is mirrored by the possible valuations: While a general variable can take the value of any object in the universe of its sort, a constructor variable can take the value of an object of the constructor sub-universe only.

General variables are the common ones in the field of term rewriting. They allow to express semantic properties that cannot be expressed by constructor variables.[31] Furthermore, the general variables allow a higher abstraction from evaluation strategies than constructor variables which result in an innermost rewriting strategy in case of free constructors.

Constructor variables are convenient in the field of inductive theorem proving[32] for expressing important lemmas that do not hold for undefined terms[33]. The means for automatically showing termination of the functions of classic inductive theorem proving (cf. Boyer&Moore[10], Walther[26]) also depend on the variables in the function definitions being bound to constructor terms only. This dependence, however, and the intended meaning of the variables at all, are usually hidden in the formalism and not made explicit as in Avenhaus&Becker[1], where it is shown that the restriction to constructor variables only, is beneficial to confluence[34] and termination of rewriting systems.

All in all, both kinds of variables have their benefits for specification with positive/negative-conditional equations and for expressing (inductive) properties with Gentzen clauses, as well as for rewriting and (inductive) theorem proving. Since the technical treatment of both kinds of variables can be achieved by simple means, we have decided to include both of them in our constructor-based approach for positive/negative-conditional equations here. Together with our generalization to positive- and negative-conditional equations, the addition of constructor variables to classic term rewriting provides us with a unifying approach to the function specification style of classic inductive theorem proving on the one hand and to term rewriting on the other.

---

[30] i. e. iff our $\overset{\circledast}{\Longrightarrow}$ is *sufficiently complete.*

[31] Consider equations for error recovery or for non-strict functions whose meaning does not depend on the definedness of all its variables, e. g. " or true $Y =$ true "

[32] Cf. Walther[27], Biundo&al.[8], Boyer&Moore[10], Zhang&al.[32], Wirth[28].

[33] E. g., one certainly should be able to express a commutativity lemma for addition of rational numbers, but one cannot expect it hold for '1/0' or other undefined terms.

[34] Cf. our Theorem 8.18

4 More Basic Notions and Notations

4.1 Substitutions

The set of *substitutions* from a variable-system $X = (X_{\varsigma,s})_{(\varsigma,s)\in\{\mathrm{SIG},\mathrm{CONS}\}\times S}$ to a $\{\mathrm{SIG},\mathrm{CONS}\}\times S$-sorted family of sets $\mathcal{T} = (\mathcal{T}_{\varsigma,s})_{(\varsigma,s)\in\{\mathrm{SIG},\mathrm{CONS}\}\times S}$ is defined to be

$$\mathcal{SUB}(X,\mathcal{T}) := \{\ \sigma : X \to \mathcal{T} \mid \forall(\varsigma,s) \in \{\mathrm{SIG},\mathrm{CONS}\}\times S : \forall x \in X_{\varsigma,s} : \sigma(x) \in \mathcal{T}_{\varsigma,s}\ \}.$$

The most important sets of substitutions are $\mathcal{SUB}(V,\mathcal{T})$ and $\mathcal{SUB}(V,\mathcal{GT})$.

For the application of the homomorphic extension of a substitution $\sigma \in \mathcal{SUB}(V,\mathcal{T})$ to a term $t \in \mathcal{T}$ we use postfix-notation: $t\sigma$. Notice that for $\sigma \in \mathcal{SUB}(V,\mathcal{GT})$: $\mathcal{V}(t\sigma) = \emptyset$ [35].

Our definition of substitutions is consistent with the notion of order-sorted substitutions if one prefers the order-sorted view[36] on our approach. A fortiori we get $\forall\sigma \in \mathcal{SUB}(V,\mathcal{T}) : \forall(\varsigma,s) \in \{\mathrm{SIG},\mathrm{CONS}\}\times S : \forall t \in \mathcal{T}_{\varsigma,s}\colon\ t\sigma \in \mathcal{T}_{\varsigma,s}$.

We use separations instead of the usual phrase "we always assume ... to have no variables in common". The set of *separations for* X, Y is defined for $X, Y \subseteq V$ by:

$$\mathrm{Sep}(X,Y) := \{\ \xi \in \mathcal{SUB}(V,V) \mid \xi \text{ bijective on } V \ \wedge\ \xi[X] \cap Y = \emptyset\ \}$$

For finite X, Y we always have $\mathrm{Sep}(X,Y) \neq \emptyset$ [37] and w. l. o. g.[38] we think '$\min \mathrm{Sep}(X,Y)$' to denote some element chosen from Sep(X, Y), because for our purposes all its elements are equivalent.

The set of *most general unifiers for E on* X is defined for $E \in \mathrm{FMul}(\mathrm{DEq}(\mathrm{sig}, V_{\mathrm{SIG}} \uplus V_{\mathrm{CONS}}))$ and $X \in \mathcal{F}(V)$ by:

$$\begin{aligned}\mathrm{Mgu}(E,X) := \{\ & \sigma \in \mathcal{SUB}(V,\mathcal{T}) \mid \mathrm{set}[E]\sigma \subseteq \mathrm{id} \ \wedge\ ^{39} \\ & \forall\mu \in \mathcal{SUB}(V,\mathcal{T}) : (\mathrm{set}[E]\mu \subseteq \mathrm{id} \ \Rightarrow\ \exists\tau \in \mathcal{SUB}(V,\mathcal{T}) : \sigma|_X\tau = \mu|_X)\ \}\end{aligned}$$

For unifiable E, i. e. for $\exists\mu \in \mathcal{SUB}(V,\mathcal{T}) : \mathrm{set}[E]\mu \subseteq \mathrm{id}$, we always have $\mathrm{Mgu}(E,X) \neq \emptyset$ and w. l. o. g. we think '$\min \mathrm{Mgu}(E,X)$' to denote some element chosen from Mgu(E, X), because for our purposes all its elements are equivalent. A unification algorithm can be found in appendix B.

---

[35] This is an application of the abuse of notation for families mentioned in sect. 2.2. A proper notation would be $\mathcal{V}(t\sigma) = (\emptyset)_{(s,\varsigma)\in\{\mathrm{SIG},\mathrm{CONS}\}\times S}$

[36] As indicated above and described in [1].

[37] since $\forall(\varsigma,s) \in \{\mathrm{SIG},\mathrm{CONS}\}\times S :\ |V_{\varsigma,s}| \notin \mathbb{N}$

[38] without loss of generality

[39] We could enlarge the conjunction by $\sigma\sigma = \sigma$ but *not*[40] by $\mathcal{V}(\sigma[\mathcal{V}(E)]) \subseteq \mathcal{V}(E)$, for which we have to choose either V_{SIG} (as in [28, 30]) or V_{CONS} (as in [1]) but cannot allow all variables of $V_{\mathrm{SIG}} \uplus V_{\mathrm{CONS}}$ in our terms.

[40] To see this, consider $x, y \in V_{\mathrm{CONS},\mathrm{nat}}$; $Y \in V_{\mathrm{SIG},\mathrm{nat}}$; a most general unifier for $\langle(x,\ \mathrm{s}\,Y)\rangle$ must be something like $\{\ x \mapsto \mathrm{s}\,y,\ Y \mapsto y\ \}$.

4.2 Algebras

We define a sig/cons-*algebra* $\mathcal{A}$ *over the signature* $\text{sig} = (\mathsf{F}, \mathsf{S}, \alpha)$ *with constructor subsignature* $\text{cons} = (\mathsf{C}, \mathsf{S}, \alpha|_{\mathsf{C}})$ to be a function $\mathcal{A} : (\mathsf{F} \uplus (\{\text{SIG}, \text{CONS}\} \times \mathsf{S})) \rightarrow \mathcal{U}$ with $\forall s \in \mathsf{S} : (\emptyset \neq \mathcal{A}(\text{CONS}, s) \subseteq \mathcal{A}(\text{SIG}, s))$ and
$f^{\mathcal{A}} : \mathcal{A}(\text{SIG}, s_0) \times \cdots \times \mathcal{A}(\text{SIG}, s_{n-1}) \rightarrow \mathcal{A}(\text{SIG}, s_n)$ for $f \in \mathsf{F}$ with $\alpha(f) = s_0 \ldots s_{n-1} s_n$,
$c^{\mathcal{A}}[\mathcal{A}(\text{CONS}, s_0) \times \cdots \times \mathcal{A}(\text{CONS}, s_{n-1})] \subseteq \mathcal{A}(\text{CONS}, s_n)$ for $c \in \mathsf{C}$ with $\alpha(c) = s_0 \ldots s_{n-1} s_n$.
We write $f^{\mathcal{A}}$ instead of $\mathcal{A}(f)$ for $f \in \mathsf{F}$. $\mathcal{A}(\varsigma, s)$ is called the *universe of* $\mathcal{A}$ *for* $(\varsigma, s) \in \{\text{SIG}, \text{CONS}\} \times \mathsf{S}$.

The sig/cons-algebras of this definition are nothing but the order-sorted algebras over the order-sorted signature exhibited in sect. 2.2.

A sig/cons-algebra $\mathcal{A}$ is called *trivial* :iff $\forall (\varsigma, s) \in \{\text{SIG}, \text{CONS}\} \times \mathsf{S} : |\mathcal{A}(\varsigma, s)| = 1$. The differences between two trivial sig/cons-algebras not being too interesting we speak of *the* trivial sig/cons-algebra if we mean any.

A (total) sig/cons-*homomorphism* $h :: \mathcal{A} \rightarrow \mathcal{B}$ from a sig/cons-algebra $\mathcal{A}$ to a sig/cons-algebra $\mathcal{B}$ is an S-sorted family $h = (h_s)_{s \in \mathsf{S}}$ of functions $h_s : \mathcal{A}(\text{SIG}, s) \rightarrow \mathcal{B}(\text{SIG}, s)$ which are compatible with sig and cons:
For $f \in \mathsf{F}$; $\alpha(f) = s_0 \ldots s_{n-1} s_n$; $\forall i < n : a_i \in \mathcal{A}(\text{SIG}, s_i)$:

$$h_{s_n}(f^{\mathcal{A}}(a_0, \ldots, a_{n-1})) = f^{\mathcal{B}}(h_{s_0}(a_0), \ldots, h_{s_{n-1}}(a_{n-1}))$$

and for all $s \in \mathsf{S}$: $\quad h_s[\mathcal{A}(\text{CONS}, s)] \subseteq \mathcal{B}(\text{CONS}, s)$

Taking the class of sig/cons-algebras for the class of objects and the class of sig/cons-homomorphisms for the class of arrows, we get the sig/cons-*homomorphism category of* sig/cons-*algebras*. The composition $hk :: \mathcal{A} \rightarrow \mathcal{C}$ of $h :: \mathcal{A} \rightarrow \mathcal{B}$ and $k :: \mathcal{B} \rightarrow \mathcal{C}$ is defined as usual by $hk := (h_s \circ k_s)_{s \in \mathsf{S}}$ and the identity homomorphism for $\mathcal{A}$ is given as $(\text{id}|_{\mathcal{A}(\text{SIG}, s)})_{s \in \mathsf{S}} :: \mathcal{A} \rightarrow \mathcal{A}$.

Let $\mathsf{X} \subseteq \mathsf{V}$. We use $\mathcal{T}(\mathsf{X})$ to denote the *term algebra over* X and sig/cons/V. This term algebra has $\mathcal{T}_{\varsigma, s} \cap \mathcal{T}(\text{sig}, \mathsf{X})$ as the universe for each $(\varsigma, s) \in \{\text{SIG}, \text{CONS}\} \times \mathsf{S}$ and works on function symbols as follows:

$$f^{\mathcal{T}(\mathsf{X})}(t_0, \ldots, t_{n-1}) = f t_0 \ldots t_{n-1} \qquad \text{for } f \in \mathsf{F} \text{ with } \alpha(f) = s_0 \ldots s_{n-1} s_n \text{ and } \forall i < n : \ t_i \in \mathcal{T}(\text{sig}, \mathsf{X})_{s_i}$$

Similarly, (by abuse of notation) we sometimes use $\mathcal{GT}$ for the *ground term algebra* $\mathcal{T}(\emptyset)$ over sig/cons instead of the family of ground terms. An $\mathcal{A}$-*valuation* κ *of* X is an element of $\mathcal{SUB}(\mathsf{X}, \mathcal{A}) := \mathcal{SUB}((\mathsf{V}_{\varsigma, s} \cap \mathsf{X})_{(\varsigma, s) \in \{\text{SIG}, \text{CONS}\} \times \mathsf{S}}, (\mathcal{A}(\varsigma, s))_{(\varsigma, s) \in \{\text{SIG}, \text{CONS}\} \times \mathsf{S}})$ [41]. The evaluation homomorphism $\mathcal{A}_\kappa :: \mathcal{T}(\mathsf{X}) \rightarrow \mathcal{A}$ is recursively defined as follows (S-index of $\mathcal{A}_\kappa$ omitted):

$$\mathcal{A}_\kappa(x) = \kappa(x) \qquad \text{for } x \in \mathsf{X}$$

$$\mathcal{A}_\kappa(f t_0 \ldots t_{n-1}) = f^{\mathcal{A}}(\mathcal{A}_\kappa(t_0), \ldots, \mathcal{A}_\kappa(t_{n-1})) \quad \text{for } f \in \mathsf{F};\ t_0, \ldots, t_{n-1} \in \mathcal{T}(\text{sig}, \mathsf{X})$$

For getting acquainted with this notation, here is a well-known lemma:

Lemma 4.1 (Substitution-Lemma)
Let $\mathcal{A}$ *be a* sig/cons-*algebra and* κ *an* $\mathcal{A}$-*valuation of* X. *For* $t \in \mathcal{T}$ *and* $\sigma \in \mathcal{SUB}(\mathsf{V}, \mathcal{T}(\mathsf{X}))$:

$$\mathcal{A}_\kappa(t\sigma) = \mathcal{A}_{\sigma \circ \mathcal{A}_\kappa}(t)$$

[41] One would not call κ a substitution, however, unless $\mathcal{A}$ is a term algebra.

For $\varsigma \in \{\mathrm{SIG}, \mathrm{CONS}\}$; dunno $\in \{\mathrm{sig}, \mathrm{cons}\}$; a sig/cons-algebra $\mathcal{A}$ is called ς:dunno-*term-generated* :iff $\forall s \in \mathrm{S} : \forall a \in \mathcal{A}(\varsigma, s) : \exists t \in \mathcal{GT}(\mathrm{dunno})_s : \mathcal{A}(t) = a$. It is called dunno-*term-generated* :iff it is SIG:dunno-term-generated.

A sig/cons-*congruence* $\sim$ *on* $\mathcal{A}$ is very similar to a sig-congruence in the multi-sorted case, namely an S-sorted family $\sim = (\sim_s)_{s\in\mathrm{S}}$ of equivalences $\sim_s$ on $\mathcal{A}(\mathrm{SIG}, s)$ being compatible with sig, i. e. satisfying for $f \in \mathsf{F}$; $\alpha(f) = s_0 \ldots s_n s_{n+1}$; $\forall i \leq n : a_i \in \mathcal{A}(\mathrm{SIG}, s_i)$:
If $a_j \sim_{s_j} b$ for some $j \leq n$, then $f^{\mathcal{A}}(a_0, \ldots, a_n) \sim_{s_{n+1}} f^{\mathcal{A}}(a_0, \ldots, a_{j-1}, b, a_{j+1}, \ldots, a_n)$.

The *factor algebra of* $\mathcal{A}$ *modulo* $\sim$ is the sig/cons-algebra $\mathcal{B}$ (denoted by $\mathcal{A}/\!\sim$) given by:

$$\mathcal{B}(\varsigma, s) := \{\ \sim_s[\{a\}] \mid a \in \mathcal{A}(\varsigma, s)\ \} \quad ((\varsigma, s) \in \{\mathrm{SIG}, \mathrm{CONS}\}\times\mathrm{S})$$

$$f^{\mathcal{B}}(\sim_{s_0}[\{a_0\}], \ldots, \sim_{s_{n-1}}[\{a_{n-1}\}]) := \sim_{s_n}[\{f^{\mathcal{A}}(a_0, \ldots, a_{n-1})\}]$$
$$(f \in \mathsf{F};\ \alpha(f) = s_0 \ldots s_{n-1} s_n;\ \forall i < n : a_i \in \mathcal{A}(\mathrm{SIG}, s_i))$$

The *canonical*[42] sig/cons-*epimorphism of* $\mathcal{A}$ *modulo* $\sim$ is the sig/cons-homomorphism $k :: \mathcal{A} \to \mathcal{A}/\!\sim$ given by ($s \in \mathrm{S}$; $a \in \mathcal{A}(\mathrm{SIG}, s)$): $k_s(a) := \sim_s[\{a\}]$.

For a sig/cons-homomorphism $h :: \mathcal{A} \to \mathcal{B}$ we define its *kernel* to be the sig/cons-congruence $\ker(h)$ given by ($s \in \mathrm{S}$; $a, b \in \mathcal{A}(\mathrm{SIG}, s)$): $(a, b) \in \ker(h)_s$:iff $h_s(a) = h_s(b)$.

The following trivial result is called a "theorem" due to its widespread area of application:

Theorem 4.2 (Homomorphism-Theorem)

Let $h :: \mathcal{A} \to \mathcal{C}$ *be a* sig/cons-*homomorphism. Let* $\sim$ *be a* sig/cons-*congruence on* $\mathcal{A}$ *with* $\forall s \in \mathrm{S}: \sim_s \subseteq \ker(h)_s$. *Define* $\mathcal{B} := \mathcal{A}/\!\sim$. *Let* k *be the canonical* sig/cons-*epimorphism of* $\mathcal{A}$ *modulo* $\sim$. *Now* $h = kl$ *uniquely defines an* S-*sorted family of functions* $l = (l_s)_{s\in\mathrm{S}}$ *with* $l_s : \mathcal{B}(\mathrm{SIG}, s) \to \mathcal{C}(\mathrm{SIG}, s)$ *for* $s \in \mathrm{S}$. *Furthermore, this* l *is a* sig/cons-*homomorphism* $l :: \mathcal{B} \to \mathcal{C}$. *Moreover, if* $\sim = \ker(h)$ *holds, then* l_s *is injective for each* $s \in \mathrm{S}$, *i. e.* $l :: \mathcal{B} \to \mathcal{C}$ *is monic in the* sig/cons-*homomorphism category of* sig/cons-*algebras.*

By specialization of notions of category theory to full sub-categories of the sig/cons-homomorphism category of sig/cons-algebras and to the forgetful functor we define for a class K of sig/cons-algebras; a sig/cons-algebra $\mathcal{A}$; X⊆V; and $\kappa \in \mathcal{SUB}(\mathrm{X}, \mathcal{A})$:
$\mathcal{A}$ is *initial in* K :iff $\mathcal{A} \in \mathrm{K}$ and for all $\mathcal{B} \in \mathrm{K}$ there is a unique $h :: \mathcal{A} \to \mathcal{B}$.
$\mathcal{A}$ is *free for* K *over* X *w. r. t.* κ :iff $\forall \mathcal{B} \in \mathrm{K} : \forall \mu \in \mathcal{SUB}(\mathrm{X}, \mathcal{B}) : \exists! h :: \mathcal{A} \to \mathcal{B} : \mu = \kappa h$.
$\mathcal{A}$ is *free in* K *over* X *w. r. t.* κ :iff $\mathcal{A} \in \mathrm{K}$ and $\mathcal{A}$ is free for K over X w. r. t. κ.

4.3 Orderings and other (binary) Relations

By an *irreflexive ordering* $<$ we mean an irreflexive and transitive relation, sometimes called "strict partial ordering" by other authors. As with all our asymmetric relation symbols: $a > b$:iff $b < a$. Speaking of an *ordering* we always think of an irreflexive ordering. A relation R is *A-reflexive* :iff $\mathrm{id}|_A \subseteq R$. Simply speaking of a *reflexive* relation we mean the biggest A that is appropriate in the local context. A *quasi-ordering* $\lesssim$ *on* A is an A-reflexive and transitive relation. An *equivalence on* A is an A-reflexive, symmetric, and transitive relation. The *equivalence* $\approx$ (on A) *of a quasi-ordering* $\lesssim$ (on A) is $\lesssim \cap \gtrsim$. The *ordering* $<$ *of a quasi-ordering* $\lesssim$ is $\lesssim \setminus \gtrsim$. A *reflexive ordering* $\leq$ *on* A is an A-reflexive, antisymmetric, and transitive relation. The *ordering* $<$ *of a reflexive ordering* $\leq$ is $\leq \setminus \mathrm{id}$. The *A-reflexive ordering* $\leq$ *of an ordering* $<$ is $< \cup\, \mathrm{id}|_A$.

[42] also called *natural* instead

Let X$\subseteq$V. A relation R on $\mathcal{T}$ is called:

X-*stable*[43] *(w. r. t. substitution)* :iff $\forall(t,t')\in R\colon \forall\sigma\in \mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{X}))\colon (t\sigma,t'\sigma)\in R$

X-*monotonic (w. r. t. replacement)* :iff
$\forall(t',t'')\in R : \forall t\in\mathcal{T}(\mathrm{sig},\mathrm{X}) : \forall p\in\mathcal{O}(t) : \forall s\in \mathrm{S} :$
$(t/p,t',t''\in\mathcal{T}_{\mathrm{SIG},s} \Rightarrow (t[p\leftarrow t'],t[p\leftarrow t''])\in R)$

sort-invariant :iff $\forall(t,t')\in R : \exists s\in\mathrm{S} :\ t,t'\in\mathcal{T}_{\mathrm{SIG},s}$

sufficiently complete (w. r. t. $\mathcal{GT}(\mathrm{cons})$*)* :iff $\forall t\in\mathcal{GT}(\mathrm{sig})\colon \exists t'\in\mathcal{GT}(\mathrm{cons})\colon (t,t')\in R$

A relation R is called:

total on A :iff $A\times A \subseteq (\mathrm{id}|_A\cup R\cup R^{-1})$
noetherian :iff there is no $a : \mathbb{N}\rightarrow \mathrm{field}(R)$ with $\forall i\in\mathbb{N} : (a_i,a_{i+1})\in R$
normalizing :iff $\forall t : \exists t' : ((t,t')\in R^{\circledast} \wedge t'\notin \mathrm{dom}(R))$

An ordering $<$ or $>$ is called *well-founded* :iff $>$ is noetherian. A quasi-ordering or a reflexive ordering is called *well-founded* :iff its ordering is well-founded.

A *reduction ordering on* $\mathcal{T}$ is a V-monotonic, V-stable, and well-founded ordering.

The *(proper) subterm ordering* $\lhd_{\mathrm{ST}}$ *on* $\mathcal{T}$ is the V-stable and well-founded ordering defined by $(t,t'\in\mathcal{T})$:

$$t\unlhd_{\mathrm{ST}}t' \text{ :iff } \exists p\in\mathcal{O}(t') : t=t'/p$$

A *simplification ordering on* $\mathcal{T}$ is a reduction ordering on $\mathcal{T}$ containing $\lhd_{\mathrm{ST}}$.

For further information on orderings see [11].

4.4 $\Longrightarrow$, Confluence, and Church-Rosser-Property

The symmetric closure of a relation $\Longrightarrow$ will be denoted by $\Longleftrightarrow$.
Two terms v, w are called *joinable w. r. t.* $\Longrightarrow$:iff $v\downarrow w$:iff $v\overset{\circledast}{\Longrightarrow}\circ\overset{\circledast}{\Longleftarrow}w$.
A relation $\Longrightarrow$ is called *confluent below u* :iff $\forall v,w : ((v\overset{\oplus}{\Longleftarrow}u\overset{\oplus}{\Longrightarrow}w) \Rightarrow (v\downarrow w))$.
A relation $\Longrightarrow$ is called *locally confluent below u* :iff $\forall v,w : ((v\Longleftarrow u\Longrightarrow w) \Rightarrow (v\downarrow w))$;
it is called *[locally] confluent* :iff it is [locally] confluent below all u.
A relation $\Longrightarrow$ is said to have the *Church-Rosser-property* :iff $\overset{\circledast}{\Longleftrightarrow}\subseteq\downarrow$.

Lemma 4.3 *A noetherian relation* $\Longrightarrow$ *is confluent* iff *it is locally confluent.*

Lemma 4.4 *A relation* $\Longrightarrow$ *is confluent* iff *it has the Church-Rosser-property.*

[43] Similarly a predicate P on $\mathcal{T}$ is X-stable :iff $\forall t\in P\colon\forall\sigma\in\mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{X}))\colon t\sigma\in P$

5 Syntax and Semantics of Specifications

Definition 5.1 (Syntax of CRS) *A (positive/negative-)*conditional rule system *(*CRS*)* R *over* sig/cons/V *is a finite subset of the* set of rules $\mathcal{RUL}(\text{sig}, \text{cons}, \text{V})$ *over* sig/cons/V, *that will be defined in Definition 6.1. The only thing we have to know about it now is:* $\mathcal{RUL}(\text{sig}, \text{cons}, \text{V}) \subseteq \text{DEq}(\text{sig}, \text{V}_{\text{SIG}} \uplus \text{V}_{\text{CONS}}) \times (\mathcal{LIT}(\text{sig}, \text{V}_{\text{SIG}} \uplus \text{V}_{\text{CONS}}))^*$, *where* $\text{DEq}(\text{sig}, \text{V}_{\text{SIG}} \uplus \text{V}_{\text{CONS}})$ *is the set of directed equations and* $\mathcal{LIT}(\text{sig}, \text{V}_{\text{SIG}} \uplus \text{V}_{\text{CONS}})$ *is the set of* condition literals *over the following predicate symbols on terms from* $\mathcal{T}(\text{sig}, \text{V}_{\text{SIG}} \uplus \text{V}_{\text{CONS}})$*:* '=', '≠' *(binary, symmetric, sort-invariant), and* 'Def' *(unary). A rule* $((l, r), \emptyset)$ *with an empty condition will be written* $l{=}r$. *A rule* $((l, r), C)$ *with condition* C *will be written* $l{=}r \longleftarrow C$. *We call* l *the* left-hand side *and* r *the* right-hand side *of the rule* $((l, r), C)$*; the terms*[44] *of the condition literals in* C *are called* condition terms *and their set is denoted by* $\mathcal{TERMS}(C)$. *The set of all left-hand sides of rules in* R *is denoted by* lhs(R). R *is* left-linear *:iff all elements of* lhs(R) *are linear terms. A rule* $((l, r), C)$ *is said to be* extra-variable free *:iff* $\mathcal{V}(r, \mathcal{TERMS}(C)) \subseteq \mathcal{V}(l)$. R *is* extra-variable free *:iff all its rules* $((l, r), C) \in \text{R}$ *are extra-variable free.*

A rule $l{=}r \longleftarrow C$ expresses a universally quantified implication with the conjunction of the literals in C as the condition and with '$l{=}r$' as the conclusion[45]. The meaning of the predicate symbols '=' and '≠' is not open to interpretation. The fixed meaning of '=' is standard; '≠' is its negation. 'Def', however, is the "definedness" predicate which states that the evaluation of its argument belongs (with sort invariant) to the *constructor sub-universe* of $\mathcal{A}$ which contains the set of evaluation values of constructor ground terms and which is intended to supply a domain for (possibly partially) defining functions on it. We speak of our new kind of model just as a "sig/cons-model" (without any further attributes), because if we removed the new predicate symbol 'Def' and the constructor sub-universes, we would just get the usual model concept of algebra; i. e., our sig/cons-model is an upward-compatible extension.

Definition 5.2 (Semantics of CRS) *Let* R *be a CRS over* sig/cons/V*; let* $\mathcal{A}$ *be a* sig/cons-*algebra. Now* $\mathcal{A}$ *is a* sig/cons-model of R **:iff**

$$\forall ((l, r), C) \in \text{R} : \forall \kappa \in \mathcal{SUB}(\text{V}, \mathcal{A}) : \Big((C \textit{ is } \textbf{true w. r. t. } \mathcal{A}_\kappa) \;\Rightarrow\; \mathcal{A}_\kappa(l) = \mathcal{A}_\kappa(r) \Big) ,$$

where C *is* **true w. r. t.** $\mathcal{A}_\kappa$ *:iff*

$$\forall s \in \text{S} : \forall u, v \in \mathcal{T}_{\text{SIG},s} : \begin{pmatrix} ((\,(u{=}v\,) \textit{ in } C) \;\Rightarrow\; \mathcal{A}_\kappa(u) = \mathcal{A}_\kappa(v)\,) & \wedge \\ ((\,(u{\neq}v\,) \textit{ in } C) \;\Rightarrow\; \mathcal{A}_\kappa(u) \neq \mathcal{A}_\kappa(v)\,) & \wedge \\ ((\,(\text{Def}\, u) \textit{ in } C) \;\Rightarrow\; \mathcal{A}_\kappa(u) \in \mathcal{A}(\text{CONS}, s)\,) & \end{pmatrix}$$

---

[44]To avoid misunderstanding: For a condition, say " $s{=}t,\ u{\neq}v,\ \text{Def}\, w$ ", we mean the top level terms $s, t, u, v, w \in \mathcal{T}(\text{sig}, \text{V}_{\text{SIG}} \uplus \text{V}_{\text{CONS}})$, but neither their proper subterms nor the literals "$s{=}t$", "$u{\neq}v$", "Def w" themselves.

[45]Notice that we don't make any use of the fact that we have $(l, r) \in \text{DEq}(\text{sig}, \text{V}_{\text{SIG}} \uplus \text{V}_{\text{CONS}})$ beyond $\langle l, r \rangle \in \text{Eq}(\text{sig}, \text{V}_{\text{SIG}} \uplus \text{V}_{\text{CONS}})$ here; this ordering-property will be used for reduction only.

As we have negative equations in our conditions, we cannot hope to get a minimum model because we can express things like ' $a{=}b \vee b{=}c$ ', which has the incomparable minimal models '$a{=}b{\neq}c$' and '$a{\neq}b{=}c$'. What we will get instead is a model that is the (up to isomorphism) uniquely determined minimum of all sig-term-generated models that are minimal w. r. t. the identification of constructor ground terms.[46] For formally expressing these minimality-properties, we need the following definition.

Definition 5.3

1. *Define* $\lesssim_{\mathrm{H}}$ *and* $\lesssim_{\mathrm{CONS}}$ *as (proper class) relations on* sig/cons-*algebras by* ($\mathcal{A}$, $\mathcal{B}$ sig/cons-*algebras):*
   $\mathcal{A} \lesssim_{\mathrm{H}} \mathcal{B}$:iff *there is a* sig/cons-*homomorphism from* $\mathcal{A}$ *to* $\mathcal{B}$.
   $\mathcal{A} \lesssim_{\mathrm{CONS}} \mathcal{B}$:iff *there is a* cons-*homomorphism from the* cons-*algebra* $\mathcal{A}|_{\mathrm{C}\uplus(\{\mathrm{CONS}\}\times\mathrm{S})}$ *to* $\mathcal{B}|_{\mathrm{C}\uplus(\{\mathrm{CONS}\}\times\mathrm{S})}$.
   We trivially get $\lesssim_{\mathrm{H}} \subseteq \lesssim_{\mathrm{CONS}}$ *(by restriction of the homomorphism); and* $\lesssim_{\mathrm{H}}$, $\lesssim_{\mathrm{CONS}}$ *are quasi-orderings. The corresponding equivalences, orderings, and reflexive orderings will be denoted by* $\approx$, $<$, $\leq$, *resp., with the corresponding subscript.*

2. *A* sig/cons-*algebra* $\mathcal{A}$ *will be called a* minimum model *(or else a* constructor-minimum model*) of a CRS* R *over* sig/cons/V :iff $\mathcal{A}$ *is a* $\lesssim_{\mathrm{H}}$*-minimum (or else* $\lesssim_{\mathrm{CONS}}$*) of the class of all* sig/cons-*models of* R.
   Similarly, a sig/cons-*algebra* $\mathcal{A}$ *will be called a* minimal model *(or else a* constructor-minimal model*) of a CRS* R *over* sig/cons/V :iff $\mathcal{A}$ *is a* sig/cons-*model of* R *and there is no* sig/cons-*model* $\mathcal{B}$ *of* R *with* $\mathcal{B} <_{\mathrm{H}} \mathcal{A}$ *(or else* $\mathcal{B} <_{\mathrm{CONS}} \mathcal{A}$ *).*

The following lemma tells us that, considering minimum models, we can think in terms of sig/cons-congruences on $\mathcal{GT}$ instead of algebras:

Lemma 5.4 *Let* $\mathcal{B}$ *be a* sig/cons-*model of the CRS* R *over* sig/cons/V. *Define the factor algebra* $\mathcal{A} := \mathcal{GT}/\mathrm{ker}(\mathcal{B})$. *Now:*

1. $\mathcal{A}$ *is a* sig/cons-*model of* R.

2. $\mathcal{A} \lesssim_{\mathrm{H}} \mathcal{B}$. *Moreover, there is a unique* sig/cons-*homomorphism* $l :: \mathcal{A} \rightarrow \mathcal{B}$ *(, which is monic in the* sig/cons-*homomorphism category of* sig/cons-*algebras).*

3. $\mathcal{A} \lesssim_{\mathrm{CONS}} \mathcal{B}$.[47]

The following lemma[48] of theoretical nature ensures the existence of minimal models:

Lemma 5.5 *Let* R *be a CRS over* sig/cons/V.

1. *The trivial* sig/cons-*algebra is a* sig/cons-*model of* R.

2. *If* $\mathcal{B}$ *is a* sig/cons-*model of* R, *then there is a minimal model* $\mathcal{A}$ *of* R *with* $\mathcal{A} \leq_{\mathrm{H}} \mathcal{B}$.

3. R *has a minimal model.*

---

[46]Cf. Corollary 6.15

[47]However, we do not have $\mathcal{A} \approx_{\mathrm{CONS}} \mathcal{B}$ in general, because $\mathcal{B}|_{\mathrm{C}\uplus(\{\mathrm{CONS}\}\times\mathrm{S})}$ need not be cons-term-generated.

[48]The lemma resembles Theorem 2.1 in Kaplan[17]. Our $\lesssim_{\mathrm{H}}$ and $\lesssim_{\mathrm{CONS}}$, however, are reflexive and therefore different from the relation $\leq$ in [17], where the homomorphism is additionally required to be unique.

6 The Reduction Relation

In this section we are going to define a reduction relation $\Longrightarrow$ which is convenient for the semantics defined in the previous section. The overall idea is to reduce a left-hand side of a rule to its right-hand side only if the condition of this rule can somehow be shown valid by means of the same reduction relation again.

Many authors impose rather strong restrictions on constructor equations, such as "no equations between constructors" ("free constructors") or "unconditional equations between constructors only". Compared to these, our restrictions are very weak. They serve to guarantee a constructor-minimum model for the constructor equations that is unique modulo $\approx_{\mathrm{CONS}}$, by requiring the constructor equations to have "Horn"-form and to be "constructor-preserving"[49].

Definition 6.1 (Set of Rules)
(continuing Definition 5.1 by adding the restrictions on constructor equations)
The set of rules *over* sig/cons/V *is defined to be:* $\mathcal{RUL}(\mathrm{sig},\mathrm{cons},\mathrm{V})$:=

$$\left\{ ((l,r),C) \in (\mathrm{DEq}(\mathrm{sig},\mathrm{V_{SIG}}\uplus\mathrm{V_{CONS}}) \times (\mathcal{LIT}(\mathrm{sig},\mathrm{V_{SIG}}\uplus\mathrm{V_{CONS}}))^*) \;\middle|\; \left(l \in \mathcal{T}(\mathrm{cons},\mathrm{V_{SIG}}\uplus\mathrm{V_{CONS}}) \Rightarrow \begin{pmatrix} \forall L \textit{ in } C : \forall u,v:\ L \neq (u{\neq}v) & \wedge \\ \mathcal{V}(r,\mathcal{TERMS}(C)) \subseteq \mathcal{V}(l) & \wedge \\ r \in \mathcal{T}(\mathrm{cons},\mathrm{V_{SIG}}\uplus\mathrm{V_{CONS}}) & \wedge \\ \mathcal{TERMS}(C) \subseteq \mathcal{T}(\mathrm{cons},\mathrm{V_{SIG}}\uplus\mathrm{V_{CONS}}) \end{pmatrix} \right) \right\}$$

We are now going to define our reduction relation, having in mind to require it to be confluent in the sequel, whereas we do not require confluence for the definition because we cannot prove confluence criteria if the non-confluent case is undefined. Therefore, we have to be explicit how we test the condition literals — even if this testing is not straightforward when confluence is not provided. Our "operational" semantics for testing condition literals is the following: '$u{=}v$' is true if u,v have reducts $\hat{u},\hat{v}$, resp., which are syntactically equal. 'Def u' is true if u has a constructor ground reduct. '$u{\neq}v$' is true if u,v have constructor ground reducts $\hat{u},\hat{v}$, resp., which are not joinable. Thus, two terms in a condition literal are "operationally" equal if they are joinable, whereas they are unequal if they are not joinable after some reduction to constructor ground terms. The non-joinability alone of two terms is not sufficient for regarding them as unequal because we are never sure about the inequality of "undefined" terms. As it often occurs, our operational logic is four-valued (i. e. '=' and '$\neq$' can independently be true or false), but in case of confluence: *tertium non datur.* In case of free or confluent constructors, the case of both '$u{=}v$' and '$u{\neq}v$' simultaneously being true means that we have something like an ambiguous function definition. Moreover, our reduction relation depends on the constructor sub-signature 'cons' beyond the signature 'sig' — just as our notion of "sig/cons-model" does.

[49]The constructor-preservation is really necessary here for guaranteeing the existence of a a minimal constructor-minimum model as in Theorem 6.14: Let $0, 1, \mathsf{true}, \mathsf{false}$ be constructor constants, let weirdp be a non-constructor constant, and take R: $1{=}0 \longleftarrow \mathsf{weirdp}{=}\mathsf{true}$; $\mathsf{weirdp}{=}\mathsf{true} \longleftarrow \mathsf{true}{\neq}\mathsf{false}$.
Now there are sig/cons-models of R with $0{\neq}1$ and models with $\mathsf{true}{\neq}\mathsf{false}$ but no models with '$0{\neq}1 \wedge \mathsf{true}{\neq}\mathsf{false}$'. Also notice, that the constructor-preservation has some additional advantages, e. g.:

1. The rules become sort-decreasing w. r. t. to the order-sorted signature exhibited in sect. 2.2, i. e. the right-hand side and the condition terms are from $\mathcal{T}(\mathrm{cons},\mathrm{V_{CONS}})$ if the left-hand side is.
2. For $u \in \mathcal{GT}$ with computable and unique normal form $\mathrm{NF}(u)$ we can test
   " $\exists \hat{u} \in \mathcal{GT}(\mathrm{cons}) : u \overset{\circledast}{\Longrightarrow} \hat{u}$ " by " $\mathrm{NF}(u) \in \mathcal{GT}(\mathrm{cons})$ ".
3. Theorem 6.16 has no reasonable analogue for CRSs which are not constructor-preserving.

Definition 6.2 (Our Reduction Relation $\Longrightarrow$)
Let R *be a CRS over* sig/cons/V. *Let* X⊆V. *The reduction relation* $\Longrightarrow_{R,X}$ *on* $\mathcal{T}(\mathrm{sig},X)$ *($\Longrightarrow$ for short) is defined to be the smallest relation satisfying the fol. requirement (:#):*
$s \Longrightarrow t$ *if* $s,t \in \mathcal{T}(\mathrm{sig},X)\ \wedge$

$$\exists((l,r),C)\in R: \exists\sigma\in\mathcal{SUB}(V,\mathcal{T}(X)): \exists p\in\mathcal{O}(s): \begin{pmatrix} s/p \;=\; l\sigma & \wedge \\ t \;=\; s[p\leftarrow r\sigma] & \wedge \\ C\sigma \text{ is } \mathbf{fulfilled\ w.\ r.\ t.} \Longrightarrow \end{pmatrix},$$

where (for $D \in \mathcal{LIT}(\mathrm{sig},X)^*$*) "D is* **fulfilled w. r. t.** $\Longrightarrow$*" is a shorthand for*

$$\forall u,v \in \mathcal{T}: \begin{pmatrix} (\ (\ (u{=}v\ )\ in\ D) & \Rightarrow & & u\downarrow v\ ) & \wedge \\ (\ (\ (\mathrm{Def}\,u)\ in\ D) & \Rightarrow & \exists\hat{u}\in\mathcal{GT}(\mathrm{cons}): & u\overset{\circledast}{\Longrightarrow}\hat{u}\ ) & \wedge \\ (\ (\ (u{\neq}v\ )\ in\ D) & \Rightarrow & \exists\hat{u},\hat{v}\in\mathcal{GT}(\mathrm{cons}): & u\overset{\circledast}{\Longrightarrow}\hat{u}\not\updownarrow\hat{v}\overset{\circledast}{\Longleftarrow}v\ )^{50} & \end{pmatrix}$$

Usually one tries to find a minimal reduction relation by taking the closure over a finitary generating relation. This is not possible here, because we have a negative condition ($\not\updownarrow$). By the "Horn"-form of our constructor equations (and the constructor-preservation), however, we know that this negative condition does not influence the reduction of constructor terms; and (in the definition) '$\not\updownarrow$' is applied to constructor (ground) terms only. Thus, we can get our minimal reduction relation by a double closure: first for constructor rules only; second for general rules knowing the constructor reduction to remain unchanged. This two step construction does not destroy the uniformity of the defining requirement(#), which allows to write uniform normal form procedures.

Define $\Longrightarrow_{R,X,0} := \emptyset$ and $\Longrightarrow_{R,X,i+1}$ to be the left-hand side[51] of the requirement(#) of Definition 6.2 with $\Longrightarrow_{R,X,i}$ substituted for $\Longrightarrow$ on the right-hand side and the additional restriction of $l \in \mathcal{T}(\mathrm{cons}, V_{\mathrm{SIG}} \uplus V_{\mathrm{CONS}})$; formally: $s \Longrightarrow_{R,X,i+1} t$:iff $s,t \in \mathcal{T}(\mathrm{sig},X)\ \wedge$

$$\exists((l,r),C)\in R: \exists\sigma\in\mathcal{SUB}(V,\mathcal{T}(X)): \exists p\in\mathcal{O}(s): \begin{pmatrix} l\in\mathcal{T}(\mathrm{cons},V_{\mathrm{SIG}}\uplus V_{\mathrm{CONS}}) & \wedge \\ s/p \;=\; l\sigma & \wedge \\ t \;=\; s[p\leftarrow r\sigma] & \wedge \\ C\sigma \text{ is } \mathbf{fulfilled\ w.\ r.\ t.} \Longrightarrow_{R,X,i} \end{pmatrix}.$$

Define $\Longrightarrow_{R,X,\omega} := \bigcup_{i\in\mathbb{N}} \Longrightarrow_{R,X,i}$ and $\Longrightarrow_{R,X,\omega+i+1}$ to be the union of $\Longrightarrow_{R,X,\omega}$ and the left-hand side[51] of the requirement(#) of Definition 6.2 with $\Longrightarrow_{R,X,\omega+i}$ substituted for $\Longrightarrow$ on the right-hand side. Finally, define $\Longrightarrow_{R,X,\omega+\omega} := \bigcup_{i\in\mathbb{N}} \Longrightarrow_{R,X,\omega+i}$. Now $\Longrightarrow_{R,X,\omega+\omega}$ satisfies the requirement(#) of Definition 6.2, and every relation satisfying this requirement must contain $\Longrightarrow_{R,X,\omega+\omega}$. Hence $\Longrightarrow_{R,X,\omega+\omega}$ is the intended smallest relation $\Longrightarrow_{R,X}$. A more detailed proof for all this can be found in appendix A. We drop "R, X" in $\Longrightarrow_{R,X}$ and $\Longrightarrow_{R,X,\beta}$ when referring to some fixed R and X. Let $\prec$ denote the ordering on the ordinal numbers. By induction over the above construction process it is trivial to verify the following corollaries.

[50]This formulation requires confluence and constructor-preservation to make sense in two-valued logic. While other formulations (e. g. a universal instead of the existential quantification) might seem to be more satisfactory, this is the one required for a correct definition. One might have expected $u\not\updownarrow v$ instead of $\exists\hat{u},\hat{v}\in\mathcal{GT}(\mathrm{cons}): u\overset{\circledast}{\Longrightarrow}\hat{u}\not\updownarrow\hat{v}\overset{\circledast}{\Longleftarrow}v$ for "$u{\neq}v$ in D" here, but this modification would not allow the conclusion that $\Longrightarrow$ is uniquely defined as can be seen from:

Example 6.3 *Let* c, d *be constructor and* a, b, e *be non-constructor constants and take* R*:* a=c ⟵ b≠d ; b=d ⟵ e≠c ; e=a . *Now* { (a, c), (e, a) } *and* { (b, d), (e, a) } *would be* ⊆*-incomparable minimal relations satisfying the modified requirement(#) of Definition 6.2. Their intersection* { (e, a) } *, however, satisfies the non-modified requirement only.*

[51]the ' if ' replaced by ' :iff '

Corollary 6.4 (Monotonicity of $\Longrightarrow$ w. r. t. Replacement)
$\Longrightarrow_{R,X,\beta}$ (for $\beta \preceq \omega{+}\omega$) and $\Longrightarrow_{R,X}$ are X-monotonic.

Corollary 6.5 (Stability of $\Longrightarrow$)
$\Longrightarrow_{R,X,\beta}$ (for $\beta \preceq \omega{+}\omega$), $\Longrightarrow_{R,X}$, and their respective fulfilledness-predicates are X-stable.

The fol. technical lemmas state constructor-preservation and that there is no need for a second closure for reduction of constructor terms.

Lemma 6.6
$\forall n \in \mathbb{N}\colon \forall s \in \mathcal{T}(\text{cons}, V_{SIG} \uplus V_{CONS})\colon \forall t\colon \Big(\ s \overset{n}{\Longrightarrow} t \ \Rightarrow\ (s \overset{n}{\Longrightarrow}_{\omega} t \in \mathcal{T}(\text{cons}, V_{SIG} \uplus V_{CONS}))\ \Big)$

Lemma 6.7 $\forall n \in \mathbb{N}\colon \forall s \in \mathcal{T}(\text{cons}, V_{CONS})\colon \forall t\colon \Big(s \overset{n}{\Longrightarrow} t \ \Rightarrow\ (s \overset{n}{\Longrightarrow}_{\omega} t \in \mathcal{T}(\text{cons}, V_{CONS}))\Big)$

Lemma 6.8 $\forall n \in \mathbb{N} : \forall s \in \mathcal{GT}(\text{cons}) : \forall t : \Big(\ s \overset{n}{\Longrightarrow} t \ \Rightarrow\ (s \overset{n}{\Longrightarrow}_{\omega} t \in \mathcal{GT}(\text{cons}))\ \Big)$

Lemma 6.9 $\downarrow \cap\ (\mathcal{T}(\text{cons}, V_{SIG} \uplus V_{CONS}) \times \mathcal{T}(\text{cons}, V_{SIG} \uplus V_{CONS}))\ \subseteq\ \downarrow_{\omega}$

Lemma 6.10 (Monotonicity of $\Longrightarrow_{\beta}$ and of Fulfilledness w. r. t. $\Longrightarrow_{\beta}$ in β)
For $\beta \preceq \gamma \preceq \omega{+}\omega$: $\Longrightarrow_{\beta}\ \subseteq\ \Longrightarrow_{\gamma}\ \subseteq\ \Longrightarrow$; and if C is fulfilled w. r. t. $\Longrightarrow_{\beta}$ and $\omega \preceq \beta\ \vee\ \forall u, v : ((u{\neq}v)$ is not in $C)$, then C is fulfilled w. r. t. $\Longrightarrow_{\gamma}$ and w. r. t. $\Longrightarrow$.

Lemma 6.11 (Fulfilledness Test may be Simple)
Let $C \in (\mathcal{LIT}(\text{sig}, X))^{*}$. If for each element $u \in \mathcal{TERMS}(C)$:

1. u has a normal form $\text{NF}(u)$ (i. e. $u \overset{*}{\Longrightarrow} \text{NF}(u) \notin \text{dom}(\Longrightarrow)$)

and

2. $\Longrightarrow$ is confluent below u,

then C is fulfilled w. r. t. $\Longrightarrow$ <u>iff</u>

$$\forall u, v \in \mathcal{T}\colon \begin{pmatrix} (\ (\ (u{=}v\ )\ \textit{in } C) \ \Rightarrow\ \text{NF}(u) = \text{NF}(v)\ ) & \wedge \\ (\ (\ (\text{Def}\, u)\ \textit{in } C) \ \Rightarrow\ \text{NF}(u) \in \mathcal{GT}(\text{cons})\ ) & \wedge \\ (\ (\ (u{\neq}v\ )\ \textit{in } C) \ \Rightarrow\ (\text{NF}(u), \text{NF}(v) \in \mathcal{GT}(\text{cons})\ \wedge\ \text{NF}(u) {\neq} \text{NF}(v))\ ) & \end{pmatrix}$$

Lemma 6.12 *Let* $X \subseteq Y \subseteq V$. *Now:*
For all $\beta \preceq \omega{+}\omega$: $\forall n \in \mathbb{N}_{+} : \overset{n}{\Longrightarrow}_{R,X,\beta}\ =\ \overset{n}{\Longrightarrow}_{R,Y,\beta} \cap (\mathcal{T}(\text{sig}, X) {\times} \mathcal{T}(\text{sig}, X))$,
and for $C \in \mathcal{LIT}(\text{sig}, X)$: C is fulfilled w. r. t. $\Longrightarrow_{R,X,\beta}$ <u>iff</u> C is fulfilled w. r. t. $\Longrightarrow_{R,Y,\beta}$.
Furthermore, $\text{dom}(\Longrightarrow_{R,X})\ =\ \text{dom}(\Longrightarrow_{R,Y}) \cap \mathcal{T}(\text{sig}, X)$.
Finally, if $\Longrightarrow_{R,Y}$ is confluent, then $\Longrightarrow_{R,X}$ is confluent, too.

By Example 6.3 we know that (for guaranteeing a constructor-minimum model) we have to restrict the terms of the negated equations to be "defined". This semantic restriction is made syntactically explicit in the fol. definition that specifies a "well-behaved" subclass of the class of CRSs, in which inequalities are founded on constructor ground terms. For a motivation cf. item (A) in sect. 3 ("Our solution"), where we discussed the problems involved.

Definition 6.13 (Def-Moderate Conditional Rule Systems (Def-MCRS))
A CRS R *is a* Def-moderate conditional rule system *(*Def-MCRS*)* <u>*:iff*</u>

$$\forall ((l, r), C) \in R : \forall (u{\neq}v)\ \textit{in } C : \Big(\ \text{Def}\, u,\ \text{Def}\, v\ \textit{are in } C\ \Big)\ .$$

Now we are able to state the fundamental theorem about $\Longrightarrow$. Its corollary says that for Def-moderate CRSs R with confluent $\Longrightarrow_{\mathrm{R},\emptyset}$, the factor algebra $\mathcal{GT}/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\emptyset}$ is an (up to isomorphism) uniquely determined sig/cons-model of R.

Theorem 6.14
(Minimal Model being Free in the Constructor-Minimal Models)
Let R *be a* Def-*MCRS over* sig/cons/V. *Let* $\mathrm{X} \subseteq \mathrm{V}$. *Let* K *be the class of all constructor-minimal*[52] *models of* R. *Let* κ *be given by* $(x \in \mathrm{X})$: $x \mapsto \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\mathrm{X}}[\{x\}]$.
Now, if $\Longrightarrow_{\mathrm{R},\emptyset}$ *is confluent*[53], *then* $\mathcal{T}(\mathrm{X})/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\mathrm{X}}$ *is free for* K *over* X *w. r. t.* κ.
Furthermore, if we assume $\Longrightarrow_{\mathrm{R},\mathrm{X}}$ *to be confluent*[54], *then the fol. items hold:*

1. $\mathcal{T}(\mathrm{X})/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\mathrm{X}}$ *is a constructor-minimum*[52] *model of* R.
2. $\mathcal{T}(\mathrm{X})/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\mathrm{X}}$ *is free in* K *over* X *w. r. t.* κ.
3. $\mathcal{T}(\mathrm{X})/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\mathrm{X}}$ *is a minimal*[52] *model of* R.

Corollary 6.15 *Let* R *be a* Def-*MCRS over* sig/cons/V. *Furthermore, assume* $\Longrightarrow_{\mathrm{R},\emptyset}$ *to be confluent. Now:* $\mathcal{GT}/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\emptyset}$ *is a minimal model of* R, *initial in the class of all constructor-minimal models of* R, *and the (up to isomorphism) unique* ($\lesssim_{\mathrm{H}}$*-) minimum of the* sig*-term-generated constructor-minimal models of* R.

Theorem 6.16
(Monotonicity of $\Longrightarrow_{\mathrm{R},\mathrm{X}}$ w. r. t. Consistent Extension of the Specification)
Let R *be a CRS over* sig/cons/V. *Let* $\mathrm{X} \subseteq \mathrm{V}$. *Let* R′ *be another CRS, but over* sig′/cons′/V′; *and* $\mathrm{X}' \subseteq \mathrm{V}'$ *with*

$$\left|\begin{array}{r} \mathrm{sig}' = (\mathsf{F}', \mathrm{S}', \alpha') \\ \mathrm{cons}' = (\mathsf{C}', \mathrm{S}', \alpha'|_{\mathsf{C}'}) \\ \mathrm{V}'|_{\{\mathrm{SIG},\mathrm{CONS}\}\times\mathrm{S}} = \mathrm{V} \\ \end{array}\right| \begin{array}{l} \mathsf{F} \subseteq \mathsf{F}' \\ \mathsf{C} \subseteq \mathsf{C}' \subseteq \mathsf{F}' \\ \mathrm{S} \subseteq \mathrm{S}' \\ \alpha \subseteq \alpha' \end{array} \left|\begin{array}{l} \mathrm{R} \subseteq \mathrm{R}' \\ \mathrm{X} \subseteq \mathrm{X}' \end{array}\right|$$

Thus, sig′/cons′/V′ *is an enrichment of* sig/cons/V *in the most general*[55] *sense we can think of. Moreover, assume*[56]*:* $\forall((l,r),C) \in (\mathrm{R}' \setminus \mathrm{R}) : \ l \notin \mathcal{T}(\mathrm{cons}, \mathrm{V}_{\mathrm{SIG}} \uplus \mathrm{V}_{\mathrm{CONS}})$ *(:\$)*
Now we have[57]*:*

1. $\forall s \in \mathcal{T}(\mathrm{cons}, \mathrm{X})\colon \forall t\colon \big((s \overset{\circledast}{\Longrightarrow}_{\mathrm{R},\mathrm{X}} t) \Leftrightarrow (s \overset{\circledast}{\Longrightarrow}_{\mathrm{R}',\mathrm{X}'} t)\big)$
   "no change on old constructor terms"
2. $\Longrightarrow_{\mathrm{R},\mathrm{X}} \subseteq \Longrightarrow_{\mathrm{R}',\mathrm{X}'}$ *"monotonicity"*
3. $\forall \beta \preceq \omega{+}\omega : \Longrightarrow_{\mathrm{R},\mathrm{X},\beta} \subseteq \Longrightarrow_{\mathrm{R}',\mathrm{X}',\beta}$ *"monotonicity"*

---

[52] Cf. Definition 5.3

[53] The remark of footnote 54 with $\mathrm{X} := \emptyset$ is applicable here.

[54] The fol. allows to apply the confluence criterion of Theorem 7.6: If we additionally require $\forall((l,r),C) \in \mathrm{R} : \forall(u{=}v)$ in C : (Def u, Def v are in C) , then we can weaken the confluence requirement to confluence of $\Longrightarrow_{\mathrm{R},\mathrm{X}} \cap (D_{\mathrm{X}} \times D_{\mathrm{X}})$ for $D_{\mathrm{X}} := \{\, u \in \mathcal{T}(\mathrm{sig}, \mathrm{X}) \mid \exists \hat{u} \in \mathcal{T}(\mathrm{cons}, \mathrm{V}_{\mathrm{CONS}} \cap \mathrm{X}) : u \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\mathrm{X}} \hat{u} \,\}$.

[55] One may even introduce new constructor symbols for the old sorts and take them from the old non-constructor symbols. Since all $\mathrm{V}_{\varsigma,s}$ are infinite, the restriction on V′ is not severe.

[56] This has to be required for keeping the negative conditions fulfilled: Having founded our inequalities on old constructor ground terms, all we have to take care of now is not to confuse these terms.

[57] While it is important that $\Longrightarrow_{\mathrm{R},\mathrm{X}}$ tests $(u{\neq}v)$ in a condition of an equation by $\exists \hat{u}, \hat{v} \in \mathcal{GT}(\underline{\underline{\mathrm{cons}}}) : u \overset{\circledast}{\Longrightarrow}_{\mathrm{R},\mathrm{X}} \hat{u} \ddagger_{\mathrm{R},\mathrm{X}} \hat{v} \overset{\circledast}{\Longleftarrow}_{\mathrm{R},\mathrm{X}} v$ instead of $\exists \hat{u}, \hat{v} \in \mathcal{GT}(\underline{\underline{\mathrm{cons}'}}) : u \overset{\circledast}{\Longrightarrow}_{\mathrm{R},\mathrm{X}} \hat{u} \ddagger_{\mathrm{R},\mathrm{X}} \hat{v} \overset{\circledast}{\Longleftarrow}_{\mathrm{R},\mathrm{X}} v$, for the validity of the theorem it does not matter whether $\Longrightarrow_{\mathrm{R},\mathrm{X}}$ is defined on $\mathcal{T}(\mathrm{sig}, \mathrm{X})$ or $\mathcal{T}(\mathrm{sig}', \mathrm{X}')$.

7 How to Test for Confluence

We are now going to define critical peaks that consist of the conditional critical pair, its peak, and the overlap position. The other notions we will use are standard, with the exception of "quasi overlay joinable" which is a slight weakening of "overlay joinable" in Dershowitz[12] (cf. below), in that it allows an identical non-overlay part in the critical pair.

Definition 7.1 (Critical Peaks and Joinability)
The set of (non-trivial) critical peaks *between two rules* $((l_k, r_k), C_k) \in \mathrm{R}$; $k < 2$; *is defined as:* $\mathrm{Cp}(\ ((l_0, r_0), C_0),\ ((l_1, r_1), C_1)\ ) :=$

$$\{\ (((l_1[p \leftarrow r_0\xi\,], r_1), C_0\xi\, C_1)\sigma,\ \ l_1\sigma,\ \ p)$$

$$\begin{array}{ll} \xi = \min \mathrm{Sep}(\mathcal{V}(((l_0, r_0), C_0)), \mathcal{V}(((l_1, r_1), C_1))) \ \wedge & \text{``no variables in common''} \\ p \in \mathcal{O}(l_1)\ \wedge\ l_1/p \notin \mathrm{V} \ \wedge & \text{``non-variable position''} \\ \sigma = \min \mathrm{Mgu}(\{(l_0\xi, l_1/p)\}, \mathcal{V}(((l_0, r_0), C_0)\xi, ((l_1, r_1), C_1))) \ \wedge & \text{``most general unifier''} \\ l_1[p \leftarrow r_0\xi\,]\sigma \neq r_1\sigma & \text{``non-trivial critical pair''} \end{array}$$

$\}$

And the set of all critical peaks of R *is* $\mathrm{CP(R)} := \bigcup_{\mathrm{rule}_0 \in \mathrm{R}} \bigcup_{\mathrm{rule}_1 \in \mathrm{R}} \mathrm{Cp}(\mathrm{rule}_0, \mathrm{rule}_1)$.

R *is said to be* overlapping *:iff* $\mathrm{CP(R)} \neq \emptyset$.

A critical peak $(((t_0, t_1), D), \hat{t}, p)$ *is* joinable w. r. t. R, X *:iff*

$$\forall \varphi \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}(\mathrm{X})) : ((D\varphi \text{ fulfilled w. r. t. } \Longrightarrow_{\mathrm{R,X}}) \Rightarrow t_0\varphi \downarrow_{\mathrm{R,X}} t_1\varphi) .$$

A critical peak $(((t_0, t_1), D), \hat{t}, p)$ *is* overlay joinable w. r. t. R, X *:iff it is joinable w. r. t.* R, X *and* $p = \emptyset$. *It is* quasi overlay joinable w. r. t. R, X *:iff*

$$\forall \varphi \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}(\mathrm{X})) \text{:} \left(\left(\begin{array}{l} D\varphi \text{ fulfilled} \\ \text{w. r. t. } \Longrightarrow_{\mathrm{R,X}} \end{array} \right) \Rightarrow \left(\begin{array}{ll} t_1\varphi = t_0\varphi[p \leftarrow t_1\varphi/p] & \wedge \\ (t_0/p)\varphi \downarrow_{\mathrm{R,X}} t_1\varphi/p \overset{\oplus}{\Longleftarrow}_{\mathrm{R,X}} (\hat{t}/p)\varphi & \end{array} \right) \right).$$

Lemma 7.2 (Joinability of Critical Peaks is Necessary for Confluence)
Let R *be a CRS over* sig/cons/V *and* X⊆V. *If* $\Longrightarrow_{\mathrm{R,X}}$ *is confluent, then all critical peaks in* CP(R) *are joinable w. r. t.* R, X.

Lemma 7.3 (Overlay Joinable ⇒ Quasi Overlay Joinable ⇒ Joinable)
Let R *be a CRS over* sig/cons/V; X⊆V; *and* $(((t_0, t_1), D), \hat{t}, p) \in \mathrm{CP(R)}$. *Now w. r. t.* R, X *the following holds:*

1. *If* $(((t_0, t_1), D), \hat{t}, p)$ *is overlay joinable, then it is quasi overlay joinable.*
2. *If* $(((t_0, t_1), D), \hat{t}, p)$ *is quasi overlay joinable, then it is joinable.*

Sufficient criteria for confluence of reduction relations for merely positive conditional rule systems are studied in Dershowitz[12]. As counterexamples for suggested sufficient confluence criteria for merely positive conditional rule systems are counterexamples for Def-MCRSs too, we repeat the results of [12] here: There are (left-linear) non-overlapping positive-conditional rule systems whose reduction relations are not (locally) confluent[58] (but necessarily non-noetherian then[59]). Therefore, syntactic confluence criteria for non-noetherian conditional rule systems must be very difficult to develop. Semantic confluence criteria (in the style of Plaisted[24]) seem to require noetherian (or at least normalizing)

[58] Cf. [12], Example A, p. 36
[59] Cf. [12], Theorem 4, p. 39

reduction relations because they rely on the irreducible reducts of the terms; furthermore irreducibility is not (semi-) decidable. Thus, for our confluence criteria we require $\Longrightarrow$ to be noetherian. Even then the situation is not very encouraging, because there are noetherian and non-confluent reduction relations of (left-linear, normal, and) positive-conditional rule systems with joinable critical peaks only.[60] Moreover, semantic confluence criteria remain difficult because irreducibility is still not (semi-) decidable. However, for merely positive conditional rule systems there are two known syntactic solutions of major[61] interest: One requires either more than joinability for the critical peaks (as, e. g., in Theorem 4 in [12]) or the condition terms to be somehow smaller than the left-hand side of the rule (as, e. g., in Theorem 3 in [12]). We will study the latter approach (which is the more important one in practice (cf. Example 7.5)) later[62]. The following result is a generalization of Theorem 4 in Dershowitz[12] from positive-conditional to positive/negative-conditional rule systems and, moreover, from overlay joinability to quasi overlay joinability.

Theorem 7.4 (Syntactic Confluence Criterion)
Let R *be a CRS over* sig/cons/V *and* X⊆V. *If* $\Longrightarrow_{\mathrm{R,X}}$ *is noetherian and all critical peaks in* CP(R) *are quasi overlay joinable w. r. t.* R, X, *then* $\Longrightarrow_{\mathrm{R,X}}$ *is confluent.*

While this theorem is very nice (theoretically) and has a pretty complicated proof, it may be difficult to apply even for merely positive conditional equations:

Example 7.5 *Let* R*:* $\mathsf{f\,s}x = \mathsf{0} \longleftarrow \mathsf{f}\,x = \mathsf{0}$; $\mathsf{f\,s}x = \mathsf{1} \longleftarrow \mathsf{f}\,x = \mathsf{1}$; f...
Assume 0 *and* 1 *to be irreducible. Now for showing the critical peak between the first two rules to be quasi overlay joinable, one has to show that it is impossible that both conditions hold simultaneously for a substitution* $\{x \mapsto t\}$. *However, in order to prove this, we need the confluence below '*f *t', which we are not allowed to assume for the joinability test here.*

Theorem 7.6 (Semantic Confluence Criterion)
Let R *be a CRS over* sig/cons/V *and* X⊆V. *Let* $\mathcal{A}$ *be a* sig/cons*-model of* R *and* κ *an* $\mathcal{A}$*-valuation of* X. *Now:*

1. *If* $\forall s \in \mathrm{S} : \forall \hat{u}, \hat{v} \in \mathcal{T}(\mathrm{sig}, \mathrm{X})_s \backslash \mathrm{dom}(\Longrightarrow_{\mathrm{R,X}}) : (\ \mathcal{A}_\kappa(\hat{u}) = \mathcal{A}_\kappa(\hat{v}) \ \Rightarrow\ \hat{u} = \hat{v}\ )$ *and* $\Longrightarrow_{\mathrm{R,X}}$ *is noetherian, then* $\Longrightarrow_{\mathrm{R,X}}$ *is confluent.*

2. *Define*[63] $D_{\mathrm{X}} := \{\ u \in \mathcal{T}(\mathrm{sig}, \mathrm{X}) \mid \exists \hat{u} \in \mathcal{T}(\mathrm{cons}, \mathrm{V_{CONS}} \cap \mathrm{X}) : u \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,X}} \hat{u}\ \}$. *If* $\forall s \in \mathrm{S} : \forall \hat{u} \in \mathcal{T}(\mathrm{cons}, \mathrm{V_{CONS}} \cap \mathrm{X})_s \backslash \mathrm{dom}(\Longrightarrow_{\mathrm{R,X}}) : \forall \hat{v} \in \mathcal{T}(\mathrm{sig}, \mathrm{X})_s \backslash \mathrm{dom}(\Longrightarrow_{\mathrm{R,X}}) :$
$$(\ \mathcal{A}_\kappa(\hat{u}) = \mathcal{A}_\kappa(\hat{v}) \ \Rightarrow\ \hat{u} = \hat{v}\ )$$
and $\Longrightarrow_{\mathrm{R,X},\omega}$ *is noetherian, then* $\Longrightarrow_{\mathrm{R,X}} \cap (D_{\mathrm{X}} \times D_{\mathrm{X}})$ *is confluent.*

---

[60] Cf. [12], Example B, p. 36

[61] We are not interested in the shallow-joinability of Dershowitz[12] here because there is a noetherian, shallow-joinable, left-linear, but not confluent, merely positive conditional rule system (cf. [12], Example C, p. 36) as well as a noetherian, shallow-joinable, normal, but not confluent, merely positive conditional rule system (cf. [12], Example D, p. 36), which means that shallow-joinability is only sufficient for confluence of rule systems which are both left-linear and normal. The combination of left-linearity and normality, however, is a too severe restriction to be of major interest for us here, because left-linearity forbids the positive part of the specification of an equality predicate by $\mathsf{eq}\,x\,x = \bot$, which is the common trick for achieving normality by transformation of '$v{=}v$' in a condition of a rule into '$\mathsf{eq}\,v\,u = \bot$'.

[62] Cf. theorems 8.17 and 8.18

[63] Cf. footnote 54

8 Compatible CRSs

Compatibility restrictions on rule systems w. r. t. well-founded orderings enhance our means of deciding reducibility and confluence. Such restrictions are necessary, as can be seen from:

Lemma 8.1 (Reducibility of Ground Terms is Not Co-semi-decidable) *There is a left-linear, non-overlapping, extra-variable free, merely positive conditional rule system* R *with noetherian and confluent reduction relation* $\Longrightarrow_{R,V}$ *for which reducibility of ground terms is not co-semi-decidable.*

Lemma 8.2 (Reducibility of Ground Terms is Not Semi-decidable) *There is a left-linear, non-overlapping, extra-variable free,* Def*-moderate CRS* R *with noetherian and confluent reduction relation* $\Longrightarrow_{R,V}$ *for which reducibility of ground terms is not semi-decidable.*

The fol. theorem is a generalization of Theorem 3.4 in Kaplan[16] in various directions.

Theorem 8.3 *Let* R *be a CRS over* sig/cons/V*. Let* X *be an enumerable subset of* V.

1. $\Longrightarrow_{R,X}$*-reducibility of terms from* $\mathcal{T}(\mathrm{sig},X)$ *is co-semi-decidable if a* $\Longrightarrow_{R,X}$*-normal form for each term from* $\mathcal{T}(\mathrm{sig},X)$ *is computable (i. e. there is a computable (partial) function* f *with* $\mathrm{dom}(f) = \{s \in \mathcal{T}(\mathrm{sig},X) \mid \exists t : (s \stackrel{\circledast}{\Longrightarrow}_{R,X} t \notin \mathrm{dom}(\Longrightarrow_{R,X}))\}$ *such that* $\forall s \in \mathrm{dom}(f) : s \stackrel{\circledast}{\Longrightarrow}_{R,X} f(s) \notin \mathrm{dom}(\Longrightarrow_{R,X})$ *).*
2. *A* $\Longrightarrow_{R,X}$*-normal form for each term from* $\mathcal{T}(\mathrm{sig},X)$ *is computable (cf. above) if* $\Longrightarrow_{R,X}$*-reducibility of terms from* $\mathcal{T}(\mathrm{sig},X)$ *is co-semi-decidable and* $\forall s \in \mathcal{GT}(\mathrm{cons}) : \exists t : s \stackrel{\circledast}{\Longrightarrow}_{R,X} t \notin \mathrm{dom}(\Longrightarrow_{R,X})$.

Corollary 8.4 *Let* R *be a CRS over* sig/cons/V*. Let* X *be an enumerable subset of* V*. Assume* $\Longrightarrow_{R,X}$ *to be noetherian. Now, co-semi-decidability of* $\Longrightarrow_{R,X}$*-reducibility of terms from* $\mathcal{T}(\mathrm{sig},X)$ *is logically equivalent to computability of a* $\Longrightarrow_{R,X}$*-normal form for each term from* $\mathcal{T}(\mathrm{sig},X)$.

8.1 The Use of Orderings

In this and the fol. section we just want to give minimal reasonable compatibility requirements for achieving additional decidability properties of our reduction relation. We start with a discussion of how to use orderings for reduction with conditional rules. This discussion mainly depends on the method of testing the conditions uniformly by the same reduction relation again, where well-founded orderings are needed for guaranteeing termination of condition-testing and reduction. Since this method does not depend on the concrete form of our rules, the situation under discussion does not differ from the case of merely positive conditional equations.

Define $s \twoheadrightarrow_{R,X} t$:iff $s \in \mathcal{T}(\mathrm{sig},X) \ \wedge\ \exists((l,r),C) \in R : \exists \sigma \in \mathcal{SUB}(V,\mathcal{T}(X)) : \exists p \in \mathcal{O}(s) :$
$(s/p = l\sigma \ \wedge\ \exists u \in \mathcal{TERMS}(C) : t = u\sigma \ \wedge\ (C\sigma \text{ is fulfilled w. r. t. } \Longrightarrow_{R,X}))$

As we test our conditions by reduction we must be allowed to switch from reduction to condition-testing, and then to reduction of the condition terms, and so on. Hence we require $(\Longrightarrow_{R,X} \cup \twoheadrightarrow_{R,X})$ to be noetherian. By the fol. lemma, this requirement can be expressed by means of relations $\rightrightarrows$, $\hookrightarrow$ on $\mathcal{T}$ as follows: $\Longrightarrow_{R,X} \subseteq \rightrightarrows$; $\twoheadrightarrow_{R,X} \subseteq \hookrightarrow$; $\rightrightarrows$ is sort-invariant, V-monotonic, and V-stable; $\hookrightarrow$ is V-stable; and $(\rightrightarrows \cup (\unrhd_{ST} \circ \hookrightarrow))$ is noetherian.

Lemma 8.5
If $(\Longrightarrow_{R,X} \cup \twoheadrightarrow_{R,X})$ *is noetherian, then* $(\Longrightarrow_{R,V} \cup (\unrhd_{ST} \circ \twoheadrightarrow_{R,V}))$ *is noetherian, too.*

The fol. two lemmas show that w. l. o. g. we can require even $(\rightrightarrows \cup \rhd_{ST} \cup \hookrightarrow)$ to be noetherian:

Lemma 8.6 *Let* $\rightrightarrows$ *be a sort-invariant*[64] *and* V*-monotonic relation on* $\mathcal{T}$. *Define* $\succ := (\rightrightarrows \cup \rhd_{ST})^{\oplus}$. *Now the following holds:*

1. *If* $\rightrightarrows$ *is noetherian [and* V*-stable], then* $\succ$ *is a well-founded [and* V*-stable] ordering, which doesn't need to be sort-invariant or* $\emptyset$*-monotonic.*
2. *(1) does not hold in general if* $\rightrightarrows$ *is not sort-invariant or not* V*-monotonic.*
3. $\rhd_{ST} \circ \rightrightarrows \;\subseteq\; \rightrightarrows \circ \rhd_{ST}$
4. $\succ \;=\; \rhd_{ST} \cup (\rightrightarrows^{\oplus} \circ \unrhd_{ST})$

Lemma 8.7 *Let*[65] $\rhd_{ST} \circ \rightrightarrows \subseteq \rightrightarrows \circ \rhd_{ST}$. *Assume* $(\rightrightarrows \cup (\unrhd_{ST} \circ \hookrightarrow))$ *to be noetherian.*[66] *Assume*[67] $(\rightrightarrows \cup \rhd_{ST})$ *to be noetherian. [Assume* $\rightrightarrows$ *and* $\hookrightarrow$ *to be* V*-stable.] Now:* $\rhd := (\rightrightarrows \cup \hookrightarrow \cup \rhd_{ST})^{\oplus}$ *is a well-founded [and* V*-stable] ordering.*

Finally, writing '$>$' for '$\rightrightarrows^{\oplus}$', this justifies the following definition:

Definition 8.8 (Termination-Pair)
A termination-pair *over* sig/V *is a pair* $(>, \rhd)$ *for which the following properties hold:*

1. $>, \rhd \;\subseteq\; \mathcal{T} \times \mathcal{T}$
2. $>$ *is a* V*-monotonic and* V*-stable*[68] *ordering*[69].
3. $\rhd$ *is a* V*-stable*[68] *and well-founded ordering.*
4. $> \subseteq \rhd$
5. $\rhd_{ST} \subseteq \rhd$ [70]

---

[64]The easiest way to achieve sort-invariance is to identify all sorts

[65]This matches Lemma 8.6(3).

[66]This matches the conclusion of Lemma 8.5.

[67]This matches the conclusion of Lemma 8.6(1).

[68]V-stability is included because it can always be achieved (for $\rightrightarrows$ and $\hookrightarrow$; and thereby for $>$ and $\rhd$, too) by restriction to ground terms — and the non-ground part of an ordering $\rhd$ whose V-stable closure is not noetherian anymore is of no use for showing termination anyway because then its $\emptyset$-stable closure is not noetherian, either.

[69]As discussed above, sort-invariance can be required here; but it is of no use for us and omitted for convenience. For the benefit from this cf. Example 8.9(3).

[70]Notice that for proving alignment (cf. Definition 8.10 below) in practice we only have to show $\rhd_{ST} \circ \rhd \subseteq \rhd$ and then take $(\rhd_{ST} \cup \rhd)^{\oplus}$ instead of $\rhd$, because then we know by Lemma 8.6 (applied to the sort-invariant restriction of $>$), and then by Lemma 8.7, that $(\rhd_{ST} \cup \rhd)^{\oplus}$ will do the job of $\rhd$.

Example 8.9 *The standard examples for a termination-pair* $(>, \rhd)$ *are:*

1. $>$ *some sort-invariant reduction ordering;* $\rhd := \rhd_{\mathrm{ST}} \cup (> \circ \unrhd_{\mathrm{ST}})$. [71]

2. $\rhd$ *some* V*-stable and well-founded ordering containing* $\rhd_{\mathrm{ST}}$*;* $> :=$ $\bigcup_{s \in \mathrm{S}} \{ (t', t'') \mid t', t'' \in \mathcal{T}_{\mathrm{SIG},s} \wedge \forall t \in \mathcal{T}\colon \forall p \in \mathcal{O}(t)\colon (t/p \in \mathcal{T}_{\mathrm{SIG},s} \Rightarrow t[p \leftarrow t'] \rhd t[p \leftarrow t'']) \}$.[72]

3. $>$ *some simplification ordering;* $\rhd := >$.

In the field of ordering restrictions for conditional rule systems the notions of "simplifying" and "reductive" are commonly used with diverse meanings. To avoid misunderstandings we use "aligned" for the local restriction on a single rule and "compatible" (with different prefixes) for the restrictions that involve the reduction relation of the whole rule system.

Definition 8.10 (Alignment of a Rule w. r. t. a Termination-Pair)
Let $(>, \rhd)$ *be a termination-pair over* sig/V.
A rule $((l,r),C) \in \mathcal{RUL}(\mathrm{sig}, \mathrm{cons}, \mathrm{V})$ *is called* aligned with $(>, \rhd)$:iff

$$l > r \quad \wedge \quad \forall u \in \mathcal{TERMS}(C):\ l \rhd u$$

Now if we compare the use of orderings in the field of conditional equations (e. g., for the "decreasing"-property in Dershowitz[13, 12]), we find that there usually is a single noetherian ordering containing $\rhd_{\mathrm{ST}}$. This ordering corresponds[73] to our ordering $\rhd$ of a termination-pair $(>, \rhd)$. Our additional $>$, however, is very useful both in practice and for establishing theoretical properties as we will see in the sequel. Furthermore, our previous discussion reveals how to establish the properties required for $\rhd$ and its interference with rules in practice.

While we require $>$ to be a reduction ordering, we avoid the superfluous commonplace restriction of $\rhd$ to be[71] $\rhd := \rhd_{\mathrm{ST}} \cup ((> \cap \bigcup_{s \in \mathrm{S}}(\mathcal{T}_{\mathrm{SIG},s} \times \mathcal{T}_{\mathrm{SIG},s})) \circ \unrhd_{\mathrm{ST}})$, because this may not be sufficient for alignment of given rules as in the fol. example of Dershowitz[74]:

Example 8.11 ($\rhd := \rhd_{\mathrm{ST}} \cup ((> \cap \bigcup_{s \in \mathrm{S}}(\mathcal{T}_{\mathrm{SIG},s} \times \mathcal{T}_{\mathrm{SIG},s})) \circ \unrhd_{\mathrm{ST}})$ **is Too Restrictive)**

$$\begin{aligned} \mathsf{b} &= \mathsf{c} \\ \mathsf{f\,b} &= \mathsf{f\,a} \\ \mathsf{a} &= \mathsf{c} \quad \longleftarrow \mathsf{b{=}c} \end{aligned}$$

Alignment of these rules requires $\mathsf{a} \rhd \mathsf{b}$ *which we cannot achieve by the above construction of* $\rhd$*:* $\mathsf{a} > \mathsf{b}$ *is impossible since alignment of the second rule requires* $\mathsf{f\,b} > \mathsf{f\,a}$ *, which also forbids* $\mathsf{a} > \mathsf{f}^{n+1}\,\mathsf{b}$ *, since then we get* $\mathsf{a} > \mathsf{f}^{(n+1)}\,\mathsf{a} > \mathsf{f}^{2(n+1)}\,\mathsf{a} > \cdots$.

Thus, for theoretical treatment, the procedure of (2) of Example 8.9 is to be preferred to that of (1) of Example 8.9, whereas (for practically guaranteeing alignment of rules) (2) of Example 8.9 lacks any hints on how to semi-decide $>$ (even for decidable $\rhd$).

All in all, there seems to be no proper reason for preferring one of $>$, $\rhd$ to the other and we thus have introduced the notion of a termination-pair $(>, \rhd)$.

---

[71] Cf. Lemma 8.6(4)

[72] While irreflexivity, transitivity, sort-invariance and V-monotonicity of $>$ are trivial, V-stability for $t' > t''$; $t', t'' \in \mathcal{T}_{\mathrm{SIG},s}$ w. r. t. a substitution $\sigma \in \mathcal{SUB}(\mathrm{V}, \mathcal{T})$ can be seen the fol. way: For arbitrary $t \in \mathcal{T}$ and $p \in \mathcal{O}(t)$ with $t/p \in \mathcal{T}_{\mathrm{SIG},s}$ define $\xi := \min \mathrm{Sep}(\mathcal{V}(t), \mathcal{V}(t', t''))$ and then $\varrho := \sigma|_{\mathcal{V}(t',t'')} \cup \xi^{-1}|_{\mathrm{V} \setminus \mathcal{V}(t',t'')}$. Now by $t\xi[p \leftarrow t'] \rhd t\xi[p \leftarrow t'']$ we get $t[p \leftarrow t'\sigma] = t\xi[p \leftarrow t']\varrho \rhd t\xi[p \leftarrow t'']\varrho = t[p \leftarrow t''\sigma]$.

[73] V-stability can always be achieved by restriction to ground terms

[74] Cf. p. 546 in [13]

8.2 Several kinds of Compatibility of CRSs

The following kind of compatibility is a generalization to negative conditions and also a slight weakening of the notion of "decreasingness" in Dershowitz[12].

Definition 8.12 (Compatibility of a CRS with a Termination-Pair)
A CRS R *over* sig/cons/V *is* X-compatible *with a termination-pair* T = (>, ⊳) *over* sig/V :iff $\forall((l,r),C) \in \mathrm{R} : \forall\tau \in \mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{X}))$:

$$\Big((C\tau \textit{ fulfilled w. r. t. } \Longrightarrow_{\mathrm{R,X}}) \;\Rightarrow\; \Big(\; ((l,r),C)\tau \textit{ is aligned with } \mathrm{T} \;\Big)\Big)$$

Compatibility of a CRS R guarantees alignment of an instantiated rule of R when its condition is fulfilled. But, while this kind of compatibility is convenient for obtaining further theoretical properties of the reduction relation, we have a problem when using this kind of compatibility of R in practice of reduction: The terms in $C\tau$ must be smaller than $l\tau$ only if $C\tau$ is fulfilled; but for easily deciding whether $C\tau$ is fulfilled we need its terms to be smaller than $l\tau$ and the analogous property for the other rules. That this need not be a vicious circle is shown by the following definition, which allows us to test the literals in the condition from left to right, using the old programming trick of a sequential "short-circuiting" AND-operator[75]. Notice that the difference to Definition 8.12 is in the quantified variable i occurring as an index which allows us to step inductively from $(< i)$ to i.

Definition 8.13 (Left-Right-Compatibility)
A CRS R *over* sig/cons/V *is* X-left-right-compatible *with a termination-pair* T = (>, ⊳) *over* sig/V :iff $\forall((l,r),L_0\ldots L_{n-1}) \in \mathrm{R} : \forall\tau \in \mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{X}))$:

$$\left(\begin{array}{lll} \forall i<n : & \big(\; ((L_0\ldots L_{i-1})\tau \textit{ fulfilled w. r. t. } \Longrightarrow_{\mathrm{R,X}}) \Rightarrow \forall u \in \mathcal{TERMS}(L_i) : l\tau \rhd u\tau & \big) \\ \wedge & \big(\; ((L_0\ldots L_{n-1})\tau \textit{ fulfilled w. r. t. } \Longrightarrow_{\mathrm{R,X}}) \Rightarrow l\tau > r\tau & \big) \end{array}\right)$$

Definition 8.14 (Don't-Care-Compatibility)
A CRS R *over* sig/cons/V *is* X-don't-care-compatible *with a termination-pair* T = (>, ⊳) *over* sig/V :iff $\forall((l,r),L_0\ldots L_{n-1}) \in \mathrm{R} : \forall\tau \in \mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{X}))$:

$$\left(\begin{array}{lll} \forall i<n : & \forall u \in \mathcal{TERMS}(L_i) : l\tau \rhd u\tau & \\ \wedge & \big(\; ((L_0\ldots L_{n-1})\tau \textit{ fulfilled w. r. t. } \Longrightarrow_{\mathrm{R,X}}) \Rightarrow l\tau > r\tau & \big) \end{array}\right)$$

Having a left-right-compatible CRS, we don't have to test the instantiated rules for alignment anymore, provided that we test the literals of the instantiated conditions from left to right until one of them fails. Having a don't-care-compatible CRS, we can even test the literals of the instantiated conditions in parallel and don't have to care for the position of these equations in the condition list. The don't-care-compatibility is conceptionally the same as the "decreasingness" in Dershowitz[12].

The "compatibility" of Definition 8.12 (which seems to be the least restrictive one tractable in theory) is intended to be an interface for generating logically stronger kinds of "compatibility" that are useful in practice (cf. definitions 8.13, 8.14), where the don't-care-compatibility seems to be the most important one. For restrictions of $\Longrightarrow_{\mathrm{R,X}}$, however, even weaker kinds of "compatibility" than the one of Definition 8.12 may be sufficient.

[75] E. g. 'and then' (instead of 'and') in SIMULA, '&&' (instead of '&') (or more accurately '_&&_' instead of '(_!=0?1:0)&(_!=0?1:0)') in C, 'AND' in LISP.

8.3 Results for Compatible CRSs

Lemma 8.15 *Let* R *be a CRS over* sig/cons/V; X⊆Y⊆V ; *and* T = (>, ⊳) *a termination-pair over* sig/V. *Assume that* R *is* Y*-compatible with* T. *Now we have* $\Longrightarrow_{\mathrm{R,Y}} \subseteq >$ *and* $\Longrightarrow_{\mathrm{R,Y}} \cup \twoheadrightarrow_{\mathrm{R,Y}} \cup \rhd_{\mathrm{ST}} \subseteq \rhd$, *which is noetherian. Furthermore,* R *is* X*-compatible with* T *and* V*-compatible with the termination-pair* ($\overset{\oplus}{\Longrightarrow}_{\mathrm{R,V}}$, $(\Longrightarrow_{\mathrm{R,V}} \cup \twoheadrightarrow_{\mathrm{R,V}} \cup \rhd_{\mathrm{ST}})^{\oplus}$) *over* sig/V.

The following notion of "weakly joinable" weakens "joinable" by adding a confluence requirement to the premise.

Definition 8.16 *A critical peak* $(((t_0,t_1),D),\hat{t},p)$ *is* ⊳-weakly joinable w. r. t. R, X *:iff* $\forall\tau \in \mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{X}))$:

$$\left(\begin{pmatrix} D\tau \textit{ fulfilled w. r. t. } \Longrightarrow_{\mathrm{R,X}} & \wedge \\ \forall u: \big(\ u \lhd \hat{t}\tau \ \Rightarrow\ (\ \Longrightarrow_{\mathrm{R,X}} \textit{ is confluent below } u\ )\big) & \end{pmatrix} \Rightarrow\ t_0\tau \downarrow_{\mathrm{R,X}} t_1\tau\right).$$

For compatible CRSs we can now give a complete confluence test à la Knuth-Bendix:

Theorem 8.17 (Syntactic Confluence Test)
Let R *be a CRS over* sig/cons/V *and* X⊆V . *Assume that* R *is* X*-compatible with a termination-pair* T = (>, ⊳) *over* sig/V. *The following two are logically equivalent:*

1. $\Longrightarrow_{\mathrm{R,X}}$ *is confluent.*
2. *All critical peaks in* CP(R) *are (*⊳*-weakly) joinable w. r. t.* R, X.

The following theorem, which is similar to Theorem 5.1 in Avenhaus&Becker[1], drops the compatibility restriction of Theorem 8.17 for those condition literals which contain constructor variables only, while it does not require (quasi) overlay joinability as Theorem 7.4.

Theorem 8.18 (Syntactic Confluence Test)
Let R *be a CRS over* sig/cons/V; X⊆V ; *and* T = (>, ⊳) *a termination-pair over* sig/V. *Assume the constructors to be free, i. e. each left-hand side of* R *contains a non-constructor symbol. Furthermore, we require the fol. compatibility-property:*
$\forall((l,r),C) \in \mathrm{R} : \forall\tau \in \mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{X}))$:

$$\left((C\tau \textit{ fulfilled w. r. t. } \Longrightarrow_{\mathrm{R,X}}) \Rightarrow \begin{pmatrix} l\tau > r\tau\ \wedge \\ \forall L \textit{ in } C : \begin{pmatrix} \forall u \in \mathcal{TERMS}(L) : l\tau \rhd u\tau \\ \vee\ \mathcal{V}(L) \subseteq \mathrm{V_{CONS}} \end{pmatrix}\end{pmatrix}\right).$$

Now, the following two are logically equivalent:

1. $\Longrightarrow_{\mathrm{R,X}}$ *is confluent.*
2. *All critical peaks in* CP(R) *are (*⊳*-weakly) joinable w. r. t.* R, X.

Lemma 8.19
Let R *be a CRS over* sig/cons/V *and* T = (>, ⊳) *a termination-pair over* sig/V. *Let* X *be an enumerable subset of* V. *Now, if*[76]

1. R *is* X*-left-right-compatible with* T,

or

2. R *is* X*-compatible with* T, $\rhd \cap (\mathcal{T}(\mathrm{sig},\mathrm{X})\times\mathcal{T}(\mathrm{sig},\mathrm{X}))$ *is semi-decidable, and* $\rhd \cap (\mathcal{GT}(\mathrm{cons})\times\mathcal{GT}(\mathrm{cons}))$ *is decidable,*

then the fol. items hold:

[76] This condition is essential: Cf. Lemma 8.2

1. $\Longrightarrow_{R,X}$*-reducibility of terms from* $\mathcal{T}(\mathrm{sig},\mathrm{X})$ *is semi-decidable.*

2. $\{\, t \mid s \overset{\oplus}{\Longrightarrow}_{R,X} t \,\}$ *is a semi-decidable set for all* $s \in \mathcal{T}(\mathrm{sig},\mathrm{X})$.

3. $\{\, t \mid s \overset{\oplus}{\Longrightarrow}_{R,X} t \,\}$ *is a finite computable set for all* $s \in \mathcal{GT}(\mathrm{cons})$.

The fol. lemma, however, shows that compatibility does not imply decidability of reducibility as long as extra-variables are permitted.

Lemma 8.20 (Reducibility of Ground Terms is Still Not Co-semi-decidable)
There is a left-linear, non-overlapping, merely positive conditional rule system R *with noetherian and confluent reduction relation* $\Longrightarrow_{R,V}$, *which is* V*-don't-care-compatible with a termination-pair* $(\rhd, \rhd)$ *with decidable* $\rhd$, *and for which reducibility of ground terms is not co-semi-decidable.*

If we do not allow extra-variables, however, we get the following decidability result, which is important due to Corollary 8.4.

Lemma 8.21
Let R *be a CRS over* sig/cons/V *and* $\mathrm{T} = (>, \rhd)$ *a termination-pair over* sig/V. *Let* X *be an enumerable subset of* V. *Now, if* R *is extra-variable free*[77] *and if*[78]

1. R *is* X*-left-right-compatible with* T,

or

2. R *is* X*-compatible with* T *and* $\rhd \cap (\mathcal{T}(\mathrm{sig},\mathrm{X}) \times \mathcal{T}(\mathrm{sig},\mathrm{X}))$ *is decidable*

then the fol. items hold:

1. $\Longrightarrow_{R,X}$*-reducibility of terms from* $\mathcal{T}(\mathrm{sig},\mathrm{X})$ *is decidable.*

2. $\{\, t \mid s \overset{\oplus}{\Longrightarrow}_{R,X} t \,\}$ *is a finite computable set for all* $s \in \mathcal{T}(\mathrm{sig},\mathrm{X})$.

3. *Confluence of* $\Longrightarrow_{R,X}$ *is co-semi-decidable.*

Confluence of $\Longrightarrow_{R,\emptyset}$ for extra-variable free, V-don't-care-compatible Def-MCRSs R cannot be semi-decidable because it is not semi-decidable even for extra-variable free, noetherian, left-linear, monadic, unconditional rule systems[79]. While confluence of $\Longrightarrow_{R,V}$, however, is decidable for noetherian, unconditional rule systems R, the fol. lemma does not give us a chance in general to decide confluence of $\Longrightarrow_{R,V}$ for extra-variable free, don't-care-compatible Def-MCRSs R.

Lemma 8.22 (Confluence of $\Longrightarrow_{R,V}$ is Not Semi-decidable)
There is a signature sig *with sub-signature* cons *and a termination-pair* $(\rhd, \rhd)$ *over* sig/V *with decidable* $\rhd$, *such that confluence of* $\Longrightarrow_{R,V}$ *is not semi-decidable in general for left-linear, extra-variable free, merely positive conditional rule systems* R *over* sig/cons/V *which are* V*-don't-care-compatible with* $(\rhd, \rhd)$.

[77]This condition is essential: Cf. Lemma 8.20
[78]This condition is essential: Cf. the lemmas 8.1 and 8.2
[79]Cf. Kapur&al.[20]

9 Inductive Validity

Definition 9.1 (Syntax of Formulas)
Let $X \subseteq V$. *The set of* formulas *(or* Gentzen clauses*)* over sig, X *is defined to be*

$$\mathrm{Form}(\mathrm{sig}, X) := \mathrm{At}(\mathrm{sig}, X)^* \times \mathrm{At}(\mathrm{sig}, X)^* ,$$

where $\mathrm{At}(\mathrm{sig}, X) \subseteq \mathcal{LIT}(\mathrm{sig}, X)$ *is the set of atoms over the following predicate symbols on terms from* $\mathcal{T}(\mathrm{sig}, X)$*: '=' (binary, symmetric, sort-invariant) and* 'Def' *(unary). A formula* (Γ, Δ) *will be written "* $\Gamma \longrightarrow \Delta$ *".*

Definition 9.2 (Validity of Formulas in sig/cons-algebras)
Let $X \subseteq V$*;* $\mathcal{A}$ *be a* sig/cons*-algebra; and* $\kappa \in \mathcal{SUB}(X, \mathcal{A})$.
An atom $(u{=}v) \in \mathrm{At}(\mathrm{sig}, X)$ *is* true w. r. t. $\mathcal{A}_\kappa$ *:iff* $\mathcal{A}_\kappa(u) = \mathcal{A}_\kappa(v)$ *;*
and an atom $(\mathrm{Def}\, u) \in \mathrm{At}(\mathrm{sig}, X)$ *(with* $u \in \mathcal{T}(\mathrm{sig}, X)_s$ *;* $s \in S$ *) is* true w. r. t. $\mathcal{A}_\kappa$ *:iff* $\mathcal{A}_\kappa(u) \in \mathcal{A}(\mathrm{CONS}, s)$.
A formula $(\Gamma, \Delta) \in \mathrm{Form}(\mathrm{sig}, X)$ *is* valid in $\mathcal{A}$ *:iff* $\forall \kappa \in \mathcal{SUB}(X, \mathcal{A})$:

$$\Big(\forall A \text{ in } \Gamma : (A \text{ is true w. r. t. } \mathcal{A}_\kappa) \;\Rightarrow\; \exists A \text{ in } \Delta : (A \text{ is true w. r. t. } \mathcal{A}_\kappa) \Big).$$

The following example illustrates why validity of a formula in all sig/cons-models may not be appropriate for our intended abstract notion of validity. Therefore, in the sequel we define and disambiguate four different conceivable abstract notions of validity, which we will call *type-A/B/C/D-inductive validity*.

Example 9.3 *(continuing Example 2.1)*
Let $x, y \in V_{\mathrm{CONS,nat}}$ *and* $l \in V_{\mathrm{CONS,list}}$. *Consider the following* Def-*MCRS:*

$$\begin{array}{llcll} \mathrm{R:} & \mathsf{memberp}\, x\, \mathsf{nil} & = & \mathsf{false} & \\ & \mathsf{memberp}\, x\, \mathsf{cons}\, y\, l & = & \mathsf{true} & \longleftarrow\;\; x = y \\ & \mathsf{memberp}\, x\, \mathsf{cons}\, y\, l & = & \mathsf{memberp}\, x\, l & \longleftarrow\;\; x \neq y\ ,\ \mathrm{Def}\, x\ ,\ \mathrm{Def}\, y \end{array}$$

Now we might have the intuition that the formula

$$\longrightarrow \mathsf{memberp}\, x\, l = \mathsf{true}\ ,\ \mathsf{memberp}\, x\, l = \mathsf{false} \qquad (:\#)$$

should be inductively valid w. r. t. to R.

But this formula is not valid in all sig/cons*-models of* R *because it is not even valid in the constructor-minimum model* $\mathcal{T}(\{l\})/\overset{\circledast}{\longleftrightarrow}_{\mathrm{R},\{l\}}$ *of* R *(cf. Theorem 6.14) (since* $\mathsf{memberp}\, 0\, l \overset{\circledast}{\not\longleftrightarrow}_{\mathrm{R},\{l\}} \mathsf{true}$ *and* $\mathsf{memberp}\, 0\, l \overset{\circledast}{\not\longleftrightarrow}_{\mathrm{R},\{l\}} \mathsf{false}$ *).*

This motivates the notions of inductive validity which are defined below and apply to this formula, in the sense that the formula (#) is type-A/B/C/D-inductively valid.

Definition 9.4 (Inductive Substitutions)
Define the set of inductive substitutions *by:*

$$\mathrm{INDSUB}(V, \mathrm{cons}) \;:=\; \{\ \tau \in \mathcal{SUB}(V, \mathcal{T}) \mid \tau[V_{\mathrm{CONS}}] \subseteq \mathcal{GT}(\mathrm{cons}) \;\wedge\; \tau|_{V_{\mathrm{SIG}}} = \mathrm{id}|_{V_{\mathrm{SIG}}}\ \}$$

Definition 9.5 (Type-A/B/C-inductive Validity)
Let R *be a CRS over* sig/cons/V*. Let* M *be the class of all* sig/cons*-models of* R*. Let* K *be the class of all constructor-minimal models of* R*. Let* $(\Gamma, \Delta) \in \mathrm{Form}(\mathrm{sig}, \mathrm{V})$*. Now* $\Gamma \longrightarrow \Delta$ *is called ...*

... type-A-inductively valid w. r. t. R <u>*:iff*</u>
$$\forall \mathcal{A} \in \mathrm{M} : \forall \tau \in \mathrm{INDSUB}(\mathrm{V}, \mathrm{cons}) : ((\Gamma, \Delta)\tau \textit{ is valid in } \mathcal{A}).$$
... type-B-inductively valid w. r. t. R <u>*:iff*</u>
$$\forall \mathcal{A} \in \mathrm{K} : \forall \tau \in \mathrm{INDSUB}(\mathrm{V}, \mathrm{cons}) : ((\Gamma, \Delta)\tau \textit{ is valid in } \mathcal{A}).$$
... type-C-inductively valid w. r. t. R <u>*:iff*</u>
$$\forall \mathcal{A} \in \mathrm{K} : \Big((\mathcal{A} \textit{ is } \mathrm{CONS{:}cons}\textit{-term-generated}) \;\Rightarrow\; ((\Gamma, \Delta) \textit{ is valid in } \mathcal{A}) \Big).$$

Type-A- and Type-B-inductive validity reduce inductive validity of formulas to validity of their inductive instances. This disallows the constructor variables of formulas to range over objects of the constructor sub-universe which are not denoted by constructor ground terms (with sort invariant). While we restrict the constructor variables in the formulas by substituting them with constructor ground terms, we do not instantiate their general variables with (general) ground terms. This is because we do not want the general variables to range over the junk generated by ground terms only, but possibly over additional junk, e. g. of non-constructor symbols which might be introduced later on. Indeed, allowing this additional junk is necessary for the monotonicity of our logic w. r. t. consistent extensions (cf. Theorem 9.17).

Type-C-inductive validity requires a model $\mathcal{A}$ considered for inductive validity to satisfy that $\mathcal{A}$ is CONS:cons-term-generated, which is equivalent to
$$\forall s \in \mathrm{S} : \mathcal{A}(\mathrm{CONS}, s) = \mathcal{A}[\mathcal{GT}(\mathrm{cons})_s] .$$
Because this means that there are no constructor objects unless they are syntactically specified by constructor ground terms (with sort invariant), this restriction can be viewed upon as another "closed world assumption" on the constructor part of the specification, besides the one on constructor ground equality used for motivating the notion of Def-moderate CRSs.

While our inductive validities of type-A/B/C do not change the "SIG-part of validity", they differ from validity in all sig/cons-models due to their influence on the "CONS-part" of the models or of the evaluation of formulas: Inductive substitutions make the "junk" of the constructor sub-universes inaccessible to constructor variables; the constructor sub-universes of CONS:cons-term-generated sig/cons-models contain no "junk" at all; and constructor-minimal models have no "confusion" in their constructor sub-universes. Thus, regarding the constructor sub-universes, we can classify type-A as "junk and confusion", type-B as "junk but no confusion", and type-C as "no junk and no confusion". À la carte, one could ask for "confusion but no junk" and define it like type-C with the 'K' replaced by an 'M', but we do not suggest this here because we think confusion to be even less digestible than junk and because understanding the dissection of the notion of inductive validity is even now difficult enough.

Before we are going to disambiguate our notions of inductive validity, we would like to emphasize that the dissection into different types of inductive validity does not require our two types of variables or the negative conditions in our rules, but instead is brought about alone by nonempty antecedents 'Γ' in formulas of the form " $\Gamma \longrightarrow \Delta$ ".

The fol. example shows the difference between type-A- and type-B-inductive validity:

Example 9.6 (Type-A $\neq$ Type-B) *(continuing Example 9.3)*

We might have the intuition that true *and* false *are not the same; i. e. that*

$$\mathsf{true} = \mathsf{false} \longrightarrow$$

should be inductively valid. This formula, however, is not valid in the trivial sig/cons-*algebra, and therefore not type-A-inductively valid (not even for* $\mathrm{R} = \emptyset$*). As the trivial* sig/cons-*algebra is not constructor-minimal, we can exclude it by requiring the models under consideration to be constructor-minimal. Indeed, the above formula is type-B-inductively valid.*

That it is very difficult to conclude from type-B- to type-A-inductive validity, can be seen from the fol. formula which is type-B- but not type-A-inductively valid (since there are sig/cons-*models of* R *which identify* s 0 *with* 0 *but do not identify* true *with* false *):*

$$\longrightarrow \mathsf{memberp\ 0\ cons\ s\ 0\ nil} = \mathsf{false}$$

The above example is characteristic for the difference between type-A- and type-B-inductive validity in the following sense:

Lemma 9.7

Let R *be a CRS over* sig/cons/V. *Let* $(\Gamma, \Delta) \in \mathrm{Form}(\mathrm{sig}, \mathrm{V})$. *Now w. r. t.* R *the fol. holds:*

1. *Type-A-inductive validity of* (Γ, Δ) *implies type-B-inductive validity of* (Γ, Δ).
2. *If no rule in* R *has a negative*[80] *condition, then type-B-inductive validity of " $\longrightarrow \Delta$ " implies type-A-inductive validity of " $\longrightarrow \Delta$ ".*

The fol. example shows the difference between type-B- and type-C-inductive validity:

Example 9.8 (Type-B $\neq$ Type-C) *(continuing Example 2.1)*

Suppose that we enrich the signature of Example 2.1 with a new non-constructor constant dunno *of sort* bool . *Since for type-B-inductive validity we also have to consider models* $\mathcal{A}$ *with* $|\mathcal{A}(\mathrm{CONS}, \mathsf{bool})| > 2$ *, the fol. formula is type-C- but not type-B-inductively valid (not even for* $\mathrm{R} = \emptyset$*):*

$$\mathrm{Def}\,\mathsf{dunno} \longrightarrow \mathsf{dunno} = \mathsf{true}\,,\ \mathsf{dunno} = \mathsf{false}\ .$$

The above example is characteristic for the difference between type-B- and type-C-inductive validity in the following sense:

Lemma 9.9

Let R *be a CRS over* sig/cons/V. *Let* $(\Gamma, \Delta) \in \mathrm{Form}(\mathrm{sig}, \mathrm{V})$. *Now w. r. t.* R *the fol. holds:*

1. *Type-B-inductive validity of* (Γ, Δ) *implies type-C-inductive validity of* (Γ, Δ).
2. *If for each atom* $(\mathrm{Def}\, u)$ *in* Γ *the formula " $\longrightarrow \mathrm{Def}\, u$ " is type-C-inductively valid, then type-C-inductive validity of* (Γ, Δ) *implies type-B-inductive validity of* (Γ, Δ).

[80] i. e. containing a literal of the form $(u \neq v)$

Example 9.10 (Type-C $\neq$ Type-D) *(continuing Example 9.3)*
Suppose that we enrich the signature of Example 9.3 with a new non-constructor symbol cdr *and add the rule* $\mathsf{cdr}\,\mathsf{cons}\,x\,l = l$ *. One might have the intuition that* $\mathsf{cdr}\,\mathsf{nil}$ *is still undefined and thus the formulas*

$$\mathsf{Def}\,\mathsf{cdr}\,\mathsf{nil} \longrightarrow$$
$$\mathsf{cdr}\,\mathsf{nil} = \mathsf{nil} \longrightarrow$$

should be inductively valid. They are not type-C-inductively valid, however, because they are not valid in the initial constructor-minimal model[81] *(cf. Corollary 6.15) of the* Def-*MCRS* R *augmented with the rule* $\mathsf{cdr}\,\mathsf{nil} = \mathsf{nil}$ *. They are type-D-inductively valid (defined below), though.*

Definition 9.11 (Type-D-inductive Validity)
Let R *be a CRS over* sig/cons/V. *Let* $(\Gamma, \Delta) \in \mathrm{Form}(\mathrm{sig}, \mathrm{V})$.
Now " $\Gamma \longrightarrow \Delta$ " *is called* type-D-inductively valid w. r. t. R *:iff*

$$(\Gamma, \Delta) \text{ is valid in } \mathcal{T}(\mathrm{V_{SIG}})/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\mathrm{V_{SIG}}} .$$

If $\Longrightarrow_{\mathrm{R},\mathrm{V_{SIG}}}$ is confluent and R is Def-moderate, then (by Theorem 6.14) $\mathcal{T}(\mathrm{V_{SIG}})/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\mathrm{V_{SIG}}}$ is a constructor-minimal model of R (i. e. $\mathcal{T}(\mathrm{V_{SIG}})/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\mathrm{V_{SIG}}} \in \mathrm{K}$) which is obviously CONS:cons-term-generated, and thus belongs to the models considered for type-C-inductive validity. Furthermore, $\mathcal{T}(\mathrm{V_{SIG}})/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\mathrm{V_{SIG}}}$ is free in K over $\mathrm{V_{SIG}}$ and therefore distinguished from the other models having no "confusion" in their constructor sub-universes by also having no confusion in its (general) universes and by all its junk being generated by $\mathrm{V_{SIG}}$.

Example 9.10 is characteristic for the difference between type-C- and type-D-inductive validity in the following sense:

Lemma 9.12
Let R *be a* Def-*MCRS over* sig/cons/V. *Let* $(\Gamma, \Delta) \in \mathrm{Form}(\mathrm{sig}, \mathrm{V})$. *Now w. r. t.* R *the following holds:*

1. *If* $\Longrightarrow_{\mathrm{R},\mathrm{V_{SIG}}}$ *is confluent*[82]*, then type-C-inductive validity of* (Γ, Δ) *implies type-D-inductive validity of* (Γ, Δ).

2. *If* $\Longrightarrow_{\mathrm{R},\emptyset}$ *is confluent*[83] *and for each* $u \in \mathcal{TERMS}(\Gamma)$ *the formula* " $\longrightarrow \mathrm{Def}\,u$ " *is type-D-inductively valid, then type-D-inductive validity of* (Γ, Δ) *implies type-C-inductive validity of* (Γ, Δ).

---

[81] which has to be considered for type-C-inductive validity because it is CONS:cons-term-generated and constructor-minimal also w. r. t. the non-augmented rule system R

[82] The following allows to apply the confluence criterion of Theorem 7.6: If we additionally require $\forall((l,r),C) \in \mathrm{R} : \forall(u{=}v)$ in C : ($\mathrm{Def}\,u$, $\mathrm{Def}\,v$ are in C) , then we can weaken the confluence requirement to confluence of $\Longrightarrow_{\mathrm{R},\mathrm{V_{SIG}}} \cap (D_{\mathrm{V_{SIG}}} \times D_{\mathrm{V_{SIG}}})$ for $D_{\mathrm{V_{SIG}}} := \{\, u \in \mathcal{T}(\mathrm{sig}, \mathrm{V_{SIG}}) \mid \exists \hat{u} \in \mathcal{GT}(\mathrm{cons}) : u \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\mathrm{V_{SIG}}} \hat{u} \,\}$.

[83] Footnote 82 is applicable here if we replace $\Longrightarrow_{\mathrm{R},\mathrm{V_{SIG}}}$ with $\Longrightarrow_{\mathrm{R},\emptyset}$ as well as $D_{\mathrm{V_{SIG}}}$ with $D_{\emptyset} := \{\, u \in \mathcal{GT}(\mathrm{sig}) \mid \exists \hat{u} \in \mathcal{GT}(\mathrm{cons}) : u \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\emptyset} \hat{u} \,\}$.

Next thing to note about type-D-inductive validity is that it is rather close to operationalization:

Lemma 9.13
Let R *be a CRS over* sig/cons/V. *Let* $(\Gamma, \Delta) \in \mathrm{Form}(\mathrm{sig}, \mathrm{V})$. *Now the following two items are logically equivalent:*

1. (Γ, Δ) *is type-D-inductively valid.*
2. $\forall \tau \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}(\mathrm{V_{SIG}}))$:

$$\left(\begin{array}{c}\Big(\forall(u{=}v) \text{ in } \Gamma\colon u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} v\tau \;\wedge\; \forall(\mathrm{Def}\, u) \text{ in } \Gamma\colon \exists \hat{u} \in \mathcal{GT}(\mathrm{cons})\colon u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} \hat{u}\Big) \\ \Rightarrow \\ \Big(\exists(u{=}v) \text{ in } \Delta\colon u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} v\tau \;\vee\; \exists(\mathrm{Def}\, u) \text{ in } \Delta\colon \exists \hat{u} \in \mathcal{GT}(\mathrm{cons})\colon u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} \hat{u}\Big)\end{array}\right)$$

By the following corollaries we see that our notions of inductive validity specialize to the single well-known notion of inductive validity of equations w. r. t. merely positive conditional specifications.

Corollary 9.14 *(of Lemma 9.7)*
Let R *be a CRS over* sig/cons/V *without negative conditions. Let* $\Delta \in \mathrm{At}(\mathrm{sig}, \mathrm{V})^*$. *Now w. r. t.* R *the following items are logically equivalent:*

1. " $\longrightarrow \Delta$ " *is type-A-inductively valid.*
2. " $\longrightarrow \Delta$ " *is type-B-inductively valid.*

Corollary 9.15 *(of Lemma 9.9)*
Let R *be a CRS over* sig/cons/V. *Let* $\Delta \in \mathrm{At}(\mathrm{sig}, \mathrm{V})^*$. *Now w. r. t.* R *the following items are logically equivalent:*

1. " $\longrightarrow \Delta$ " *is type-B-inductively valid.*
2. " $\longrightarrow \Delta$ " *is type-C-inductively valid.*

Corollary 9.16 *(of Lemma 9.12)*
Let R *be a* Def-*MCRS over* sig/cons/V. *Let* $\Delta \in \mathrm{At}(\mathrm{sig}, \mathrm{V})^*$. *Furthermore, assume* $\Longrightarrow_{\mathrm{R,V_{SIG}}}$ *to be confluent. Now w. r. t.* R *the following items are logically equivalent:*

1. " $\longrightarrow \Delta$ " *is type-C-inductively valid.*
2. " $\longrightarrow \Delta$ " *is type-D-inductively valid.*

Having established our notions of inductive validity, let us now have a brief look on notions of inductive validity in literature, most of which can be described as specializations of our notions of inductive validity. If we consider all symbols to be constructor symbols (and, a fortiori, drop our negative conditions in our rules), we find type-A-inductive validity in Kounalis&Rusinowitch[22] as well as in Bouhoula&al.[9]. The notion of inductive validity of Zhang&al.[32] can be described as type-A when we implicitly take all variables in rules for general variables and all variables in formulas for constructor variables. Considering again all symbols to be constructor symbols (but allowing an unrestricted set of Gentzen clauses over equality atoms for specification)[84], type-A-inductive validity had

[84]instead of our R, which is restricted to be a subset of our set of rules $\mathcal{RUL}(\mathrm{sig}, \mathrm{cons}, \mathrm{V})$

also been included in a preliminary version of Bachmair&Ganzinger[3]. In the final version, however, after developing their *perfect* model in Bachmair&Ganzinger[4], they have dropped it; instead, we now find in Ganzinger&Stuber[14] inductive validity defined to be validity in the perfect model,[85] which is more similar to type-D with our free constructor-minimal model (which their approach (allowing more general specifications) does not provide) replaced with the perfect model. In Padawitz[23] and Kounalis&Rusinowitch[21] (contrary to Kounalis&Rusinowitch[22] (cf. above)), we find the usual validity in the initial model which is like[86] our type-D. Bevers&Lewi[7] also use initial model or type-D-inductive validity (without non-constructor symbols and without negative conditions in rules or formulas); however, they claim it to be equivalent to type-A, which is obviously not correct, even in their restricted context. We have already discussed the general ideas of Kapur&Musser[18, 19] in sect. 3 ("Our Solution"; around footnote 24). We have no notion of inductive validity corresponding to their notion in [18]. However, if a constructor-minimum model $\mathcal{A}$ exists and we take a choice set of $\mathcal{GT}(\mathrm{cons})/(\mathrm{ker}(\mathcal{A}) \cap (\mathcal{GT}(\mathrm{cons}) \times \mathcal{GT}(\mathrm{cons})))$ for their set of constructor terms, the notion of inductive validity of [19] coincides[86] (at least for their restriction to formulas of the form " $\longrightarrow u = v$ ") with our type-C and (in the absence of 'Def'-literals equivalently) type-B.

Now it is difficult to conclude which notion of inductive validity is the appropriate one. We think that all our four types are of interest. Type-C and type-D are our favourites because they capture the intuition of our approach. (Reconsider the formulas of Example 9.6 as well as the formula of Example 9.8.) In particular, the permission of confusion in the constructor sub-universes of the models considered for type-A, conflicts with our treatment of negative conditions of rules by negation as failure restricted to the objects of the constructor sub-universes, which was justified in sect. 3 by a closed world assumption on equality of these objects. The second formula of Example 9.6 illustrates this conflict: Since we have " $\mathsf{memberp}\,\mathsf{0}\,\mathsf{cons}\,\mathsf{s}\,\mathsf{0}\,\mathsf{nil} \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\emptyset} \mathsf{false}$ " we really would like " $\longrightarrow \mathsf{memberp}\,\mathsf{0}\,\mathsf{cons}\,\mathsf{s}\,\mathsf{0}\,\mathsf{nil} = \mathsf{false}$ " to be inductively valid. A more objective criterion for convenience of a notion of inductive validity is its suitability for theorem proving. Type-D is rather close to operationalization (cf. Lemma 9.13) and a prover for it may be used (via Lemma 9.12(2)) for showing type-C. This procedure is not complete (cf. Example 9.8) for establishing type-C-inductive validity; but the type-C-inductively valid formulas we lose seem to be in general very difficult to prove. While the details of inductive theorem proving methods and techniques within our constructor-based approach are far beyond the scope of this paper, one important feature which a notion of inductive validity needs for being convenient for inductive theorem proving is its monotonicity w. r. t. consistent extension of the specification: Contrary to deductive first order theorem proving, inductive theorem proving often is only successful when one tries to show stronger theorems than one initially intended to show. This is because induction hypotheses are not only a task but also a tool for the inductive argumentation. Sometimes the required induction hypotheses or lemmas are not expressible by our formulas unless we extend the specification in a consistent manner. Consistent extensions also play an important role for incremental refinement and modular construction of specifications. Since then we do not want to lose the theorems already shown, we need some means to tell whether they are still valid in the extended specification. Thus a notion of inductive validity can be said to be more adequate the

[85] still considering all symbols to be constructor symbols and allowing an unrestricted set of Gentzen clauses over equality atoms for specification

[86] when we consider all symbols to be constructor symbols and drop our negative conditions in our rules

more monotonic w. r. t. consistent extension of the specification it is. Luckily, (contrary to perfect model validity) our four notions behave rather well:

Theorem 9.17 (Monotonicity of Inductive Validity of Formulas w. r. t. Consistent Extension of the Specification)

Let R *be a CRS over* sig/cons/V. *Let* R′ *be another CRS, but over* sig′/cons′/V′; *with*

$$\left|\begin{array}{rl} \mathrm{sig}' &= (\mathrm{F}',\mathrm{S}',\alpha') \\ \mathrm{cons}' &= (\mathrm{C}',\mathrm{S}',\alpha'|_{\mathrm{C}'}) \\ \mathrm{V}'|_{\{\mathrm{SIG},\mathrm{CONS}\}\times\mathrm{S}} &= \mathrm{V} \\ & \end{array}\right. \left|\begin{array}{l} \mathrm{F} \subseteq \mathrm{F}' \\ \mathrm{C} \subseteq \mathrm{C}' \subseteq \mathrm{F}' \\ \mathrm{S} \subseteq \mathrm{S}' \\ \alpha \subseteq \alpha' \end{array}\right| \begin{array}{l} \mathrm{R} \subseteq \mathrm{R}' \,| \\ \\ \\ \\ \end{array}$$

and for[87] $c \in \mathrm{C}'\backslash\mathrm{C}\ ;\ \alpha'(c) = s_0 \ldots s_{n-1} \longrightarrow s_n\ :\ \ s_n \in \mathrm{S}'\backslash\mathrm{S}$.
Thus, sig′/cons′/V′ *is an enrichment of* sig/cons/V *in a very general*[88] *sense.*

Moreover, assume[89]*:* $\forall((l,r),C) \in \mathrm{R}' \setminus \mathrm{R} : l \notin \mathcal{T}(\mathrm{cons}, \mathrm{V}_{\mathrm{SIG}} \uplus \mathrm{V}_{\mathrm{CONS}})$. *(:$1)*
Assume[90] $\Longrightarrow_{\mathrm{R}',\emptyset}$ *to be confluent*[91]. *(:$2)*

Now, for $(\Gamma,\Delta) \in \mathrm{Form}(\mathrm{sig},\mathrm{V})$, *we have:*

A. *(Even without the assumptions ($1), ($2):) If* (Γ,Δ) *is type-A-inductively valid w. r. t.* R, *then* (Γ,Δ) *is type-A-inductively valid w. r. t.* R′, *too.*

B. *If* R′ *is* Def*-moderate, and if* (Γ,Δ) *is type-B-inductively valid w. r. t.* R, *then* (Γ,Δ) *is type-B-inductively valid w. r. t.* R′, *too.*

C. *If* R′ *is* Def*-moderate, and if* (Γ,Δ) *is type-C-inductively valid w. r. t.* R, *then* (Γ,Δ) *is type-C-inductively valid w. r. t.* R′, *too.*

D. *If for each* $u \in \mathcal{TERMS}(\Gamma)$ *the formula "* $\longrightarrow \mathrm{Def}\, u$ *" is type-D-inductively valid w. r. t.* R,[92] *and if* (Γ,Δ) *is type-D-inductively valid w. r. t.* R, *then* (Γ,Δ) *is type-D-inductively valid w. r. t.* R′, *too.*

[87]If we did not require this, we would get new constructor ground terms of old sorts and new inductive instances of old formulas, which clearly destroys type-A/B/C/D-inductive validity. E. g.: If we add the new constructor constant strange–list of sort list to the specification of Example 9.3, then the formula (#) becomes type-A/B/C/D-inductively invalid.

[88]Please notice that the last requirement is the only additional one on the signatures compared to Theorem 6.16.

[89]This has to be required for keeping the negative conditions of the instantiated rules being fulfilled: Having founded our inequalities on old constructor ground terms, all we have to take care of now is not to confuse these terms.

[90]Confluence is necessary of course: Consider the rule system

$$\mathsf{nicep} = \mathsf{true} \longleftarrow \mathsf{true} \neq \mathsf{false}\ ,\ \mathrm{Def}\,\mathsf{false}\ ,\ \mathrm{Def}\,\mathsf{true}$$

and the type-B/C/D-inductively valid formulas $\mathsf{true} = \mathsf{false} \longrightarrow$

$$\longrightarrow \mathsf{nicep} = \mathsf{true}\ ,$$

none of which keeps being type-B/C/D-inductively valid if we add the rules $\mathsf{dunno} = \mathsf{true}$
$\mathsf{dunno} = \mathsf{false}$.

[91]The following allows to apply the confluence criterion of Theorem 7.6: If we additionally require $\forall((l,r),C) \in \mathrm{R}' : \forall(u{=}v)$ in C : (Def u, Def v are in C) , then we can weaken the confluence requirement to confluence of $\Longrightarrow_{\mathrm{R}',\emptyset} \cap (D'_\emptyset \times D'_\emptyset)$ for $D'_\emptyset := \{\, u \in \mathcal{GT}(\mathrm{sig}') \mid \exists \hat{u} \in \mathcal{GT}(\mathrm{cons}') : u \overset{*}{\Longleftrightarrow}_{\mathrm{R}',\emptyset} \hat{u} \,\}$.

[92]This means that each $\mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{V}_{\mathrm{SIG}}))$-instance of a term occurring in the antecedent Γ must be defined, i. e. have a congruent constructor ground term. This restriction is necessary because R′ may define some terms that were undefined in R: In the situation of Example 9.10 we can destroy the type-D-inductive validity of the two formulas there by adding the rule cdr nil = nil .

10 Conclusion

We have presented a novel constructor-based approach to positive/negative-conditional equational specifications, which was heavily inspired by previous work of Kapur&Musser[18, 19] (for the case of unconditional equations only) and Zhang[31]. Under some reasonable syntactical restrictions on the form of positive/negative-conditional rules it turns out that the combination of these ideas with the approach of Kaplan[17] becomes very fruitful and also relevant for practical purposes since many natural specifications involve both conditional equations with positive and negative conditions and partially defined functions. For such specifications we have been able to define semantics admitting a unique model, being initial in the class of constructor-minimal models, if [ground] confluence of our reduction relation is provided. The lack of an initial model was one of the main disadvantages of the approach of Kaplan[17]. The addition of constructor variables conceptually completes the constructor-based approach and shows up new possibilities for the practice of specification. Furthermore, a thorough and precise analysis of termination and decidability issues has led to some useful and slightly weakened "decreasingness"-notions for positive/negative-conditional rule systems. Moreover, we have also been able to provide some interesting confluence criteria. Finally, we have defined and disambiguated several notions of inductive validity in our constructor-based approach. Since (under reasonable assumptions) all these kinds of validity are monotonic w. r. t. consistent extension of the specification, the whole approach may be considered to be a firm theoretical basis for (first-order) inductive theorem proving in theories specified by positive/negative-conditional equations.

Dedication:
To my father, Friedhelm Wirth, in gratefulness.

Acknowledgements:
We would like to thank Jürgen Avenhaus and Klaus Becker for fruitful discussions, and Klaus Madlener and Rüdiger Lunde for useful hints.

Editorial Remark:
This report is a thorough revision of Wirth&Gramlich[29][93] including lots of significant refinements and extensions[94].

[93] Cf. also the shortened version Wirth&Gramlich[30], which has been published in the proceedings of the Third International Workshop on Conditional Term Rewriting Systems in Pont-à-Mousson, July 1992.

[94] Besides lots of important details, Wirth&Gramlich[29] lacks the following: Constructor variables, embedding into the order-sorted framework, parameterization of $\Longrightarrow_{\mathrm{R,X}}$ in X, Theorem 6.14, Theorem 7.6, Lemma 8.5, Theorem 8.18 and the rest of sect. 8.3, and (most important) the entire sect. 9 about inductive validity.

A The Proofs

Proof of Lemma 5.4

1. Due to $\ker(\mathcal{A}) = \ker(\mathcal{B})$, $\mathcal{A}$ and $\mathcal{B}$ do not differ in their evaluation of two ground terms of the same sort regarding '=' or '≠'. Furthermore, if $\mathcal{A}(t) \in \mathcal{A}(\mathrm{CONS}, s)$ for some $t \in \mathcal{GT}_{\mathrm{SIG},s}$, then there is some $t' \in \mathcal{GT}_{\mathrm{CONS},s}$ with $\mathcal{B}(t') = \mathcal{B}(t)$, by which we get $\mathcal{B}(t) \in \mathcal{B}(\mathrm{CONS}, s)$. Thus 'Def t' is true w. r. t. $\mathcal{A}$ only if it is true w. r. t. $\mathcal{B}$. Since 'Def' occurs as a positive literal in the conditions of the rules only, $\mathcal{A}$ is a sig/cons-model of each ground rule of which $\mathcal{B}$ is a sig/cons-model.
   Now, if $\mathcal{B}$ is a sig/cons-model of R, by the Substitution-Lemma(4.1), it is a sig/cons-model of all ground instances of the rules of R, which implies that $\mathcal{A}$ is a sig/cons-model of all ground instances of the rules of R, which again implies that $\mathcal{A}$ is a sig/cons-model of all rules of R. This last step can be seen the following way: Let κ be any $\mathcal{A}$-valuation of V. Then by the Axiom of Choice there is a $\sigma \in \mathcal{SUB}(\mathrm{V}, \mathcal{GT})$ such that $\sigma \circ \mathcal{A} = \kappa$. Then by the Substitution-Lemma $\forall t \in \mathcal{T} : \mathcal{A}_\kappa(t) = \mathcal{A}(t\sigma)$.
2. This is nothing but an application of the Homomorphism-Theorem(4.2).
3. By (*2*) and $\lesssim_{\mathrm{H}} \subseteq \lesssim_{\mathrm{CONS}}$.

Proof of Lemma 5.5

1. Trivial.
2. If $\mathcal{B}$ is minimal itself, we are finished. If not, by Lemma 5.4(2) we can w. l. o. g. assume $\mathcal{B}$ to be a factor algebra of $\mathcal{GT}$ and consider congruences on $\mathcal{GT}$ instead of sig/cons-algebras. We only have to apply Zorn's Lemma to get a minimal congruence whose factor algebra is a sig/cons-model of R, because this factor algebra will be a minimal sig/cons-model of R. Therefore, we are now going to show that the premise of Zorn's Lemma is satisfied.

   Let $\mathrm{CHAIN} = \{\sim_i \mid i \in I\}$ be a nonempty $\subseteq$-chain of congruences on $\mathcal{GT}$ with $\forall i \in I : (\mathcal{GT}/\!\sim_i$ is a sig/cons-model of R) . Define $\approx := \bigcap_{i\in I} \sim_i$. Of course, $\approx$ is a congruence on $\mathcal{GT}$ that bounds CHAIN below. The only thing left to be shown is that $\mathcal{GT}/\!\approx$ is a sig/cons-model of R.
   Contrariwise there was[95] a $(l{=}r \longleftarrow C) \in \mathrm{R}$ and a $\sigma \in \mathcal{SUB}(\mathrm{V}, \mathcal{GT})$ with
   $\forall L \text{ in } C : \forall u,v : (L = (u{=}v) \Rightarrow u\sigma \approx v\sigma)$;
   $\forall L \text{ in } C : \forall u : (L = (\mathrm{Def}\, u) \Rightarrow \exists \hat{u} \in \mathcal{GT}(\mathrm{cons}) : u\sigma \approx \hat{u})$;
   $\forall L \text{ in } C : \forall u,v : (L = (u{\neq}v) \Rightarrow u\sigma \not\approx v\sigma)$; $l\sigma \not\approx r\sigma$.
   Now, by asking for the "reasons" of the '$\not\approx$', there is a J with $J \subseteq I$;
   $\forall L \text{ in } C : \forall u,v : (L = (u{\neq}v) \Rightarrow \exists j_L \in J : u\sigma \not\sim_{j_L} v\sigma)$;
   $\exists j_{(l,r)} \in J : l\sigma \not\sim_{j_{(l,r)}} r\sigma$; and $|J| \leq |\{\, L \text{ in } C \mid \exists u,v : L = (u{\neq}v) \,\}| + 1$.
   Define $\equiv := \bigcap_{j\in J} \sim_j$. As $\approx \subseteq \equiv$, $\equiv$ yields no model of R for the same "reason" as $\approx$. But (as J finite) $\{\sim_j \mid j \in J\}$ has a $\subseteq$-minimal[96] element $\sim_{j_0}$ with $j_0 \in J$. Now $\equiv = \sim_{j_0}$. Hence $\sim_{j_0}$ yields no sig/cons-model of R, which is a contradiction.
3. By (*1*) and (*2*).

[95] We tacitly use the Substitution-Lemma(4.1) again, just the way we used it at the end of (1) in the proof of Lemma 5.4.
[96] and minimum

Proof of Correctness of Definition 6.2

Let $\Longrightarrow$ be the intersection of all relations satisfying the requirement of Definition 6.2. We claim the following:

1. $\Longrightarrow_{\mathrm{R,X},\omega}$ is the minimum of all relations satisfying the requirement of Definition 6.2 with the additional restriction of $l \in \mathcal{T}(\mathrm{cons}, \mathrm{V_{SIG}} \uplus \mathrm{V_{CONS}})$.

2. $\forall s \in \mathcal{T}(\mathrm{cons}, \mathrm{V_{SIG}} \uplus \mathrm{V_{CONS}}) : \forall t : (s \overset{\circledast}{\Longrightarrow}_{\mathrm{R,X},\omega} t \quad \Rightarrow \quad t \in \mathcal{T}(\mathrm{cons}, \mathrm{V_{SIG}} \uplus \mathrm{V_{CONS}}))$

3. $\forall i \in \mathrm{I\!N} : \forall n \in \mathrm{I\!N} : \; (\overset{n}{\Longrightarrow}_{\mathrm{R,X},\omega+i} \cap (\mathcal{T}(\mathrm{cons}, \mathrm{V_{SIG}} \uplus \mathrm{V_{CONS}}) \times \mathcal{T})) \subseteq \overset{n}{\Longrightarrow}_{\mathrm{R,X},\omega}$

4. $\forall i \in \mathrm{I\!N} : \; \Longrightarrow_{\mathrm{R,X},\omega+i} \; \subseteq \; \Longrightarrow_{\mathrm{R,X},\omega+i+1}$

5. $\Longrightarrow_{\mathrm{R,X}}$ satisfies the requirement of Definition 6.2; i. e. $\Longrightarrow \, \subseteq \, \Longrightarrow_{\mathrm{R,X}}$.

6. $\Longrightarrow_{\mathrm{R,X},\omega} \; \subseteq \; \Longrightarrow$

7. $(\Longrightarrow \cap (\mathcal{T}(\mathrm{cons}, \mathrm{V_{SIG}} \uplus \mathrm{V_{CONS}}) \times \mathcal{T})) \; \subseteq \; \Longrightarrow_{\mathrm{R,X},\omega}$

8. $\forall i \in \mathrm{I\!N} : \; \Longrightarrow_{\mathrm{R,X},\omega+i} \; \subseteq \; \Longrightarrow$

9. $\Longrightarrow_{\mathrm{R,X}}$ is the minimum of all relations which satisfy the requirement of Definition 6.2.

To the proofs of these claims:

1. By the restriction on the rules of Definition 6.1, $\Longrightarrow_{\mathrm{R,X},\omega}$ is just the standard closure over a finitary relation.

2. By the restriction on the rules of Definition 6.1.

3. By induction on i using (1), (2), and the restriction on the rules of Definition 6.1.

4. By induction on i using (3) and the restriction on the rules of Definition 6.1.

5. By (4), taking $\Longrightarrow_{\mathrm{R,X},\omega+i+1}$ on the left-hand side of the requirement for the maximum i of all $\Longrightarrow_{\mathrm{R,X},\omega+i}$ occurring positively (i. e. not in a $\not\updownarrow$-statement) on the right-hand side of the requirement.

6. By (1), $\Longrightarrow_{\mathrm{R,X},\omega}$ is the intersection of a superset of the set whose intersection is $\Longrightarrow$.

7. By (3) and (5).

8. <u>$i = 0$:</u> By (6). <u>$i \Rightarrow (i+1)$:</u> By (6), (7), and (2).

9. By (5) and (8).

<u>Q. e. d. (Correctness Proof for Definition 6.2)</u>

Proof of Lemma 6.6
By (9), (3), and (2) in the correctness proof for Definition 6.2 above.
Q. e. d. (Lemma 6.6)

Proof of Lemma 6.7
By the restriction on the rules of Definition 6.1 and induction over i one gets
$\forall i \in \mathbb{N} : \forall n \in \mathbb{N} : \forall s \in \mathcal{T}(\text{cons}, \text{V}_{\text{CONS}}) : \forall t : (\ s \overset{n}{\Longrightarrow}_{\text{R,X},i} t \ \Rightarrow\ t \in \mathcal{T}(\text{cons}, \text{V}_{\text{CONS}})\ )$.
By Lemma 6.6 this is sufficient. Q. e. d. (Lemma 6.7)

Proof of Lemma 6.8
By the restriction on the rules of Definition 6.1 and induction over i one gets
$\forall i \in \mathbb{N} : \forall n \in \mathbb{N} : \forall s \in \mathcal{GT}(\text{cons}) : \forall t : (\ s \overset{n}{\Longrightarrow}_{\text{R,X},i} t \ \Rightarrow\ t \in \mathcal{GT}(\text{cons})\ )$.
By Lemma 6.6 this is sufficient. Q. e. d. (Lemma 6.8)

Proof of Lemma 6.9
By Lemma 6.6. Q. e. d. (Lemma 6.9)

Proof of Lemma 6.10
By (4), (9) in the correctness proof above we get $\Longrightarrow_{\text{R,X},\beta} \subseteq \Longrightarrow_{\text{R,X},\gamma} \subseteq \Longrightarrow$. A literal L in C (being fulfilled w. r. t. $\Longrightarrow_{\text{R,X},\beta}$) must have one of the following forms: If $L = (u{=}v)$, then by $u{\downarrow}_{\text{R,X},\beta} v$ we get $u{\downarrow}_{\text{R,X},\gamma} v$. If $L = (\text{Def}\, u)$, then there is some $\hat{u} \in \mathcal{GT}(\text{cons})$ with $u \overset{\circledast}{\Longrightarrow}_{\text{R,X},\beta} \hat{u}$ and we get $u \overset{\circledast}{\Longrightarrow}_{\text{R,X},\gamma} \hat{u}$. If $L = (u{\neq}v)$, then there are $\hat{u}, \hat{v} \in \mathcal{GT}(\text{cons})$ with $u \overset{\circledast}{\Longrightarrow}_{\text{R,X},\beta} \hat{u} \updownarrow_{\text{R,X},\beta} \hat{v} \overset{\circledast}{\Longleftarrow}_{\text{R,X},\beta} v$ and by $\omega \preceq \beta$ we get $u \overset{\circledast}{\Longrightarrow}_{\text{R,X},\gamma} \hat{u} \updownarrow_{\text{R,X},\omega} \hat{v} \overset{\circledast}{\Longleftarrow}_{\text{R,X},\gamma} v$ and finally by Lemma 6.9 $u \overset{\circledast}{\Longrightarrow}_{\text{R,X},\gamma} \hat{u} \updownarrow_{\text{R,X},\gamma} \hat{v} \overset{\circledast}{\Longleftarrow}_{\text{R,X},\gamma} v$. Q. e. d. (Lemma 6.10)

Proof of Lemma 6.11
The 'if'-part of the proof is trivial. The 'only if'-part is straightforward as follows: If $u \downarrow v$, then by confluence (below v) $u \overset{\circledast}{\Longrightarrow} \text{NF}(v)$ and then by confluence (below u) $\text{NF}(u) = \text{NF}(v)$. If $u \overset{\circledast}{\Longrightarrow} \hat{u} \in \mathcal{GT}(\text{cons})$ then by confluence $\hat{u} \overset{\circledast}{\Longrightarrow} \text{NF}(u)$ and then by Lemma 6.8 $\text{NF}(u) \in \mathcal{GT}(\text{cons})$. If $\hat{u}, \hat{v} \in \mathcal{GT}(\text{cons})$ and $u \overset{\circledast}{\Longrightarrow} \hat{u} \updownarrow \hat{v} \overset{\circledast}{\Longleftarrow} v$ then by confluence $\text{NF}(u) \overset{\circledast}{\Longleftarrow} \hat{u} \updownarrow \hat{v} \overset{\circledast}{\Longrightarrow} \text{NF}(v)$; hence $\text{NF}(u) \neq \text{NF}(v)$ and by Lemma 6.8 $\text{NF}(u), \text{NF}(v) \in \mathcal{GT}(\text{cons})$. Q. e. d. (Lemma 6.11)

Proof of Lemma 6.12
By the Axiom of Choice there is some $\tau \in \mathcal{SUB}(\text{Y}, \mathcal{T}(\text{X}))$ with $\tau|_{\text{X}} \subseteq \text{id}$. Using this τ in combination with Corollary 6.5 for getting rid of variables from Y\X (introduced by extra-variables), the first sentence is trivial by induction on β. The rest then follows immediately from Corollary 6.5. Q. e. d. (Lemma 6.12)

Proof of Theorem 6.14
Let $\mathcal{A} := \mathcal{T}(\text{X}) / \overset{\circledast}{\Longleftrightarrow}_{\text{R,X}}$. Let $\mathcal{I} := \mathcal{GT} / \overset{\circledast}{\Longleftrightarrow}_{\text{R},\emptyset}$.
Claim 1: If $\mathcal{C}$ is a sig/cons-model of R; $\mu \in \mathcal{SUB}(\text{X}, \mathcal{C})$; then

$$\forall \beta \preceq \omega : \forall s \in \text{S} : \ \Longrightarrow_{\text{R,X},\beta} \cap\, (\mathcal{T}_{\text{SIG},s} \times \mathcal{T}_{\text{SIG},s}) \ \subseteq\ \ker(\mathcal{C}_\mu)_s \ .$$

The proof of Claim 1 is omitted because it reads just like the proof of Claim 2 until it comes to $\beta \succ \omega$.
Proof of (*1.*): By the Axiom of Choice, each element of $\mathcal{SUB}(\text{V}, \mathcal{A})$ can be written $\sigma \circ \mathcal{A}_\kappa$ for some $\sigma \in \mathcal{SUB}(\text{V}, \mathcal{T}(\text{X}))$. Thus (by the Substitution-Lemma(4.1)), for $\mathcal{A}$ being a sig/cons-model of R it is sufficient to note that (by confluence of $\Longrightarrow_{\text{R,X}}[\cap(D_{\text{X}} \times D_{\text{X}})]$, the fact that R is a Def-MCRS, and Lemma 6.8) for $((l,r),C) \in \text{R}$; $\sigma \in \mathcal{SUB}(\text{V}, \mathcal{T}(\text{X}))$:

$C\sigma$ is fulfilled w. r. t. $\Longrightarrow_{R,X}$ iff

$$\forall u,v \in \mathcal{T}: \left(\begin{array}{lll} (\,(\,(u{=}v\,) \text{ in } C\sigma) & \Rightarrow & u \overset{\circledast}{\Longleftrightarrow}_{R,X} v\,) \;\wedge \\ (\,(\,(\mathrm{Def}\,u) \text{ in } C\sigma) & \Rightarrow & \exists \hat{u} \in \mathcal{GT}(\mathrm{cons}) : u \overset{\circledast}{\Longleftrightarrow}_{R,X} \hat{u}\,) \;\wedge \\ (\,(\,(u{\neq}v\,) \text{ in } C\sigma) & \Rightarrow & u \overset{\circledast}{\not\Longleftrightarrow}_{R,X} v\,) \end{array}\right)$$

For the proof of $\mathcal{A}$ being a constructor-minimum model, suppose $\mathcal{C}$ to be a sig/cons-model of R. We have to find a cons-homomorphism from $\mathcal{A}|_{\mathrm{C}\uplus(\{\mathrm{CONS}\}\times\mathrm{S})}$ to $\mathcal{C}|_{\mathrm{C}\uplus(\{\mathrm{CONS}\}\times\mathrm{S})}$. Let $\mathcal{B}$ be the term algebra over $\mathrm{V_{CONS}} \cap \mathrm{X}$ and cons/$\mathrm{V_{CONS}}$. There is a cons-homomorphism $h :: \mathcal{A}|_{\mathrm{C}\uplus(\{\mathrm{CONS}\}\times\mathrm{S})} \to (\mathcal{B}/(\overset{\circledast}{\Longleftrightarrow}_{R,X} \cap (\mathcal{T}(\mathrm{cons},\mathrm{V_{CONS}})\times\mathcal{T}(\mathrm{cons},\mathrm{V_{CONS}}))))$ given by ($s \in \mathrm{S}$; $A \in \mathcal{A}(\mathrm{CONS},s)$): $A \mapsto \mathcal{T}(\mathrm{cons},\mathrm{V_{CONS}}) \cap A$. Thus we only have to find a cons-homomorphism from $\mathcal{B}/(\overset{\circledast}{\Longleftrightarrow}_{R,X} \cap (\mathcal{T}(\mathrm{cons},\mathrm{V_{CONS}})\times\mathcal{T}(\mathrm{cons},\mathrm{V_{CONS}})))$ to $\mathcal{C}|_{\mathrm{C}\uplus(\{\mathrm{CONS}\}\times\mathrm{S})}$. Let μ be a $\mathcal{C}$-valuation of X (which always exists by the Axiom of Choice). Using the Homomorphism-Theorem (the usual one for cons-homomorphisms, not ours) all we have to show is $\forall s \in \mathrm{S} : \overset{\circledast}{\Longleftrightarrow}_{R,X} \cap (\mathcal{T}_{\mathrm{CONS},s}\times\mathcal{T}_{\mathrm{CONS},s}) \subseteq \ker(\mathcal{C}_\mu)_s$, which by confluence of $\Longrightarrow_{R,X}[\cap(D_X\times D_X)]$ is the same as $\forall s \in \mathrm{S} : \downarrow_{R,X} \cap (\mathcal{T}_{\mathrm{CONS},s}\times\mathcal{T}_{\mathrm{CONS},s}) \subseteq \ker(\mathcal{C}_\mu)_s$. Because of Lemma 6.7, $\forall s \in \mathrm{S} : \Longrightarrow_{R,X,\omega} \cap (\mathcal{T}_{\mathrm{CONS},s}\times\mathcal{T}_{\mathrm{CONS},s}) \subseteq \ker(\mathcal{C}_\mu)_s$ is sufficient for this. But this is implied by Claim 1. Q. e. d. (*1.*)

Claim 2: If $\mathcal{C} \in \mathrm{K}$; $\mu \in \mathcal{SUB}(\mathrm{X},\mathcal{C})$; $\Longrightarrow_{R,\emptyset}[\cap(D_\emptyset\times D_\emptyset)]$ is confluent; then

$$\forall \beta \preceq \omega{+}\omega : \forall s \in \mathrm{S} : \Longrightarrow_{R,X,\beta} \cap (\mathcal{T}_{\mathrm{SIG},s}\times\mathcal{T}_{\mathrm{SIG},s}) \subseteq \ker(\mathcal{C}_\mu)_s .$$

Proof of Claim 2: For the non-limit ordinals 0, ω, $\omega{+}\omega$ the induction step is trivial. For a non-limit ordinal $\beta{+}1$ the induction step is as follows: Suppose $s \Longrightarrow_{R,X,\beta+1} t$. If $\omega \preceq \beta$ and $s \Longrightarrow_{R,X,\omega} t$, then we already have $\mathcal{C}_\mu(s) = \mathcal{C}_\mu(t)$. Otherwise there are $((l,r),C) \in \mathrm{R}$; $\sigma \in \mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{X}))$; $p \in \mathcal{O}(s)$; with $s/p = l\sigma$; $t = s[p \leftarrow r\sigma]$; and $C\sigma$ fulfilled w. r. t. $\Longrightarrow_{R,X,\beta}$. As $\mathcal{C}$ is a sig/cons-model of R, for $\mathcal{C}_\mu(s) = \mathcal{C}_\mu(t)$ (by the Substitution-Lemma(4.1)) we only have to show that the condition $C\sigma$ is true w. r. t. $\mathcal{C}_\mu$. Three cases for L in $C\sigma$: If $L = (u{=}v)$, by $u \downarrow_{R,X,\beta} v$ the induction hypothesis implies $\mathcal{C}_\mu(u) = \mathcal{C}_\mu(v)$. If $L = (\mathrm{Def}\,u)$ with $u \in \mathcal{T}_{\mathrm{SIG},s'}$, by the existence of some $\hat{u} \in \mathcal{GT}_{\mathrm{CONS},s'}$ with $u \overset{\circledast}{\Longrightarrow}_{R,X,\beta} \hat{u}$ the induction hypothesis implies $\mathcal{C}_\mu(u) = \mathcal{C}_\mu(\hat{u}) \in \mathcal{C}(\mathrm{CONS},s')$. If $L = (u{\neq}v)$, by the existence of some $\hat{u},\hat{v} \in \mathcal{GT}(\mathrm{cons})$ with $u \overset{\circledast}{\Longrightarrow}_{R,X,\beta} \hat{u} \not\downarrow_{R,X,\beta} \hat{v} \overset{\circledast}{\Longleftarrow}_{R,X,\beta} v$ the induction hypothesis implies $\mathcal{C}_\mu(u) = \mathcal{C}_\mu(\hat{u}) = \mathcal{C}(\hat{u})$ and $\mathcal{C}(\hat{v}) = \mathcal{C}_\mu(\hat{v}) = \mathcal{C}_\mu(v)$. Thus, for $\mathcal{C}_\mu(u) \neq \mathcal{C}_\mu(v)$ it is sufficient to show $\mathcal{C}(\hat{u}) \neq \mathcal{C}(\hat{v})$. By $\omega \preceq \beta$ and Lemma 6.9 we know that $\hat{u} \not\downarrow_{R,X} \hat{v}$; then by Lemma 6.12 $\hat{u} \not\downarrow_{R,\emptyset} \hat{v}$; and therefore (by confluence of $\Longrightarrow_{R,\emptyset}[\cap(D_\emptyset\times D_\emptyset)]$) $\hat{u} \overset{\circledast}{\not\Longleftrightarrow}_{R,\emptyset} \hat{v}$. Because of (*1*) (for $\mathrm{X} := \emptyset$), $\mathcal{C}$ must be not only a constructor-minimal model but also a constructor-minimum model of R. Hence $\mathcal{C}(\hat{u}) \neq \mathcal{C}(\hat{v})$ simply because $\hat{u},\hat{v} \in \mathcal{GT}(\mathrm{cons})$; $\mathcal{I}(\hat{u}) \neq \mathcal{I}(\hat{v})$; and $\mathcal{I}$ is (by (*1*)) a sig/cons-model of R. Q. e. d. (Claim 2)

Claim 3: If $\Longrightarrow_{R,\emptyset}[\cap(D_\emptyset\times D_\emptyset)]$ is confluent, then $\mathcal{A}$ is free for K over X w. r. t. κ.

Proof of Claim 3: Suppose $\mathcal{C} \in \mathrm{K}$ and μ to be a $\mathcal{C}$-valuation of X. The uniqueness of the required sig/cons-homomorphism $h :: \mathcal{A} \to \mathcal{C}$ with $\mu = \kappa h$ is trivial. For its existence (by the Homomorphism-Theorem(4.2)) we only have to show

$$\forall s \in \mathrm{S} : \overset{\circledast}{\Longleftrightarrow}_{R,X} \cap (\mathcal{T}_{\mathrm{SIG},s}\times\mathcal{T}_{\mathrm{SIG},s}) \subseteq \ker(\mathcal{C}_\mu)_s ,$$

which is implied by Claim 2. Q. e. d. (Claim 3)

Claim 4: If $\Longrightarrow_{R,X}[\cap(D_X\times D_X)]$ is confluent, then $\Longrightarrow_{R,\emptyset}[\cap(D_\emptyset\times D_\emptyset)]$ is confluent, too.

Proof of Claim 4: Trivial by Lemma 6.12. Q. e. d. (Claim 4)

Proof of (*2.*): By (*1*) and the claims 4 and 3.

Proof of (*3.*): Suppose $\mathcal{C}$ to be a sig/cons-model of R with $\mathcal{C} \lesssim_{\mathrm{H}} \mathcal{A}$. By (*1*) we get $\mathcal{C} \in \mathrm{K}$, and then by (*2*) $\mathcal{A} \lesssim_{\mathrm{H}} \mathcal{C}$.

Proof of Theorem 6.16

First note that the remark of footnote 57 is respected during the whole proof.

Claim 1: $\forall i \in \mathbb{N}$: $\forall n \in \mathbb{N}$: $\forall s \in \mathcal{T}(\mathrm{cons}, \mathrm{X})$: $\forall t$:

$$\left(s \overset{n}{\Longrightarrow}_{\mathrm{R'},\mathrm{X'},i} t \;\Rightarrow\; (s \overset{n}{\Longrightarrow}_{\mathrm{R},\mathrm{X},i} t \in \mathcal{T}(\mathrm{cons}, \mathrm{X}))\right)$$

Claim 2: $\forall \beta \preceq \omega{+}\omega : \Longrightarrow_{\mathrm{R},\mathrm{X},\beta} \subseteq \Longrightarrow_{\mathrm{R'},\mathrm{X'},\beta}$

Claim 2 and Claim 1 (using Lemma 6.6 and $\mathcal{T}(\mathrm{cons}, \mathrm{V_{SIG}} \uplus \mathrm{V_{CONS}}) \subseteq \mathcal{T}(\mathrm{cons'}, \mathrm{V'_{SIG}} \uplus \mathrm{V'_{CONS}})$) imply (*1*); and Claim 2 implies (*2*) and (*3*).

Proof of Claim 1: $i = 0$: $\Longrightarrow_{\mathrm{R'},\mathrm{X'},0} = \emptyset$. $i \Rightarrow (i+1)$: $n = 0$: Trivial.

$n \Rightarrow (n+1)$: Suppose $s \overset{n}{\Longrightarrow}_{\mathrm{R'},\mathrm{X'},i+1} s' \Longrightarrow_{\mathrm{R'},\mathrm{X'},i+1} t$. By induction hypothesis in n we know $s \overset{n}{\Longrightarrow}_{\mathrm{R},\mathrm{X},i+1} s' \in \mathcal{T}(\mathrm{cons}, \mathrm{X})$. By ($) there must be some $l \in \mathcal{T}(\mathrm{cons}, \mathrm{V_{SIG}} \uplus \mathrm{V_{CONS}})$; $((l,r),C) \in \mathrm{R}$; $\sigma \in \mathcal{SUB}(\mathcal{V}(l), \mathcal{T})$; $p \in \mathcal{O}(s')$ with $s'/p = l\sigma$; $t = s'[p \leftarrow r\sigma]$; $C\sigma$ fulfilled w. r. t. $\Longrightarrow_{\mathrm{R'},\mathrm{X'},i}$; $\mathrm{ran}(\sigma) \subseteq \mathcal{T}(\mathrm{cons}, \mathrm{X})$. By the structure of constructor equations we have no inequality literal in $C\sigma$. For $(u{=}v)$ in $C\sigma$ we have $u, v \in \mathcal{T}(\mathrm{cons}, \mathrm{X})$ and $u \downarrow_{\mathrm{R'},\mathrm{X'},i} v$ and therefore by induction hypothesis in i: $u \downarrow_{\mathrm{R},\mathrm{X},i} v$. For $(\mathrm{Def}\, u)$ in $C\sigma$ we have $u \in \mathcal{T}(\mathrm{cons}, \mathrm{X})$ and $u \overset{\circledast}{\Longrightarrow}_{\mathrm{R'},\mathrm{X'},i} \hat{u}$ for some $\hat{u} \in \mathcal{GT}(\mathrm{cons'})$ and hence by induction hypothesis in i: $u \overset{\circledast}{\Longrightarrow}_{\mathrm{R},\mathrm{X},i} \hat{u} \in (\mathcal{T}(\mathrm{cons}, \mathrm{X}) \cap \mathcal{GT}(\mathrm{cons'})) = \mathcal{GT}(\mathrm{cons})$. Thus, finally we conclude that $C\sigma$ is fulfilled w. r. t. $\Longrightarrow_{\mathrm{R},\mathrm{X},i}$, i. e. $s' \Longrightarrow_{\mathrm{R},\mathrm{X},i+1} t \in \mathcal{T}(\mathrm{cons}, \mathrm{X})$. Q. e. d. (Claim 1)

Proof of Claim 2: The induction step for the limit ordinals 0, ω, and $\omega + \omega$ is trivial. For a non-limit ordinal $\beta + 1$ the induction step is as follows: Suppose $s \Longrightarrow_{\mathrm{R},\mathrm{X},\beta+1} t$. In the case of $\omega \preceq \beta$ this may be due to $s \Longrightarrow_{\mathrm{R},\mathrm{X},\omega} t$; but then by induction hypothesis we succeed by $\Longrightarrow_{\mathrm{R},\mathrm{X},\omega} \subseteq \Longrightarrow_{\mathrm{R'},\mathrm{X'},\omega} \subseteq \Longrightarrow_{\mathrm{R'},\mathrm{X'},\beta+1}$. Otherwise there must be some $((l,r),C) \in \mathrm{R}$; $\sigma \in \mathcal{SUB}(\mathrm{V}, \mathcal{T})$; $p \in \mathcal{O}(s)$ with $s/p = l\sigma$; $t = s[p \leftarrow r\sigma]$; $C\sigma$ fulfilled w. r. t. $\Longrightarrow_{\mathrm{R},\mathrm{X},\beta}$. The only thing to be shown is: $C\sigma$ fulfilled w. r. t. $\Longrightarrow_{\mathrm{R'},\mathrm{X'},\beta}$. For $(u{=}v)$ in $C\sigma$ we have $u \downarrow_{\mathrm{R},\mathrm{X},\beta} v$ and therefore by induction hypothesis $u \downarrow_{\mathrm{R'},\mathrm{X'},\beta} v$. For $(\mathrm{Def}\, u)$ in $C\sigma$ we have $u \overset{\circledast}{\Longrightarrow}_{\mathrm{R},\mathrm{X},\beta} \hat{u}$ for some $\hat{u} \in \mathcal{GT}(\mathrm{cons})$ and therefore by induction hypothesis $u \overset{\circledast}{\Longrightarrow}_{\mathrm{R'},\mathrm{X'},\beta} \hat{u}$. For $(u{\neq}v)$ in $C\sigma$ we have $u \overset{\circledast}{\Longrightarrow}_{\mathrm{R},\mathrm{X},\beta} \hat{u} \not\downarrow_{\mathrm{R},\mathrm{X},\beta} \hat{v} \overset{\circledast}{\Longleftarrow}_{\mathrm{R},\mathrm{X},\beta} v$ for some $\hat{u}, \hat{v} \in \mathcal{GT}(\underline{\mathrm{cons}})$ and $\omega \preceq \beta$. We get $\hat{u} \not\downarrow_{\mathrm{R},\mathrm{X},\omega} \hat{v}$, and then (by Claim 1) $\hat{u} \not\downarrow_{\mathrm{R'},\mathrm{X'},\omega} \hat{v}$ and (by Lemma 6.9) $\hat{u} \not\downarrow_{\mathrm{R'},\mathrm{X'}} \hat{v}$. Finally, by induction hypothesis we get $u \overset{\circledast}{\Longrightarrow}_{\mathrm{R'},\mathrm{X'},\beta} \hat{u} \not\downarrow_{\mathrm{R'},\mathrm{X'},\beta} \hat{v} \overset{\circledast}{\Longleftarrow}_{\mathrm{R'},\mathrm{X'},\beta} v$.

Proof of Lemma 7.2

For $(((t_0,t_1),D),\hat{t},p) \in \mathrm{CP}(\mathrm{R})$ there are $((l_k,r_k),C_k) \in \mathrm{R}$; $k < 2$; $\xi, \sigma \in \mathcal{SUB}(\mathrm{V}, \mathcal{T})$ with $l_0\xi\sigma = l_1\sigma/p$; $((t_0,t_1),D) = ((l_1[p \leftarrow r_0\xi], r_1), C_0\xi\, C_1)\sigma$. Let $\varphi \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}(\mathrm{X}))$ and assume $D\varphi$ to be fulfilled. Then $C_0\xi\sigma\varphi$ and $C_1\sigma\varphi$ are fulfilled, too. Therefore: $t_0\varphi = l_1\sigma\varphi[p \leftarrow r_0\xi\sigma\varphi] \Longleftarrow l_1\sigma\varphi \Longrightarrow r_1\sigma\varphi = t_1\varphi$ and by confluence we have $t_0\varphi \downarrow t_1\varphi$. Q. e. d. (Lemma 7.2)

Proof of Lemma 7.3

(*1*): For $p = \emptyset$ the definition of "quasi overlay joinable" reads:

$$\forall \varphi \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}(\mathrm{X})) : (\; D\varphi \text{ fulfilled } \Rightarrow\; t_0\varphi \downarrow t_1\varphi \overset{\oplus}{\Longleftarrow} \hat{t}\varphi)$$

As we assume $(((t_0,t_1),D),\hat{t},\emptyset)$ to be joinable, we now only have to show $t_1\varphi \overset{\oplus}{\Longleftarrow} \hat{t}\varphi$ under the assumption of $D\varphi$ being fulfilled. There are $((l_k,r_k),C_k) \in \mathrm{R}$; $k < 2$; $\xi, \sigma \in \mathcal{SUB}(\mathrm{V}, \mathcal{T})$ with $(((\ldots,t_1),D),\hat{t}) = (((\ldots,r_1),C_0\xi\, C_1),l_1)\sigma$. Since $C_1\sigma\varphi$ is fulfilled, we have $t_1\varphi = r_1\sigma\varphi \Longleftarrow l_1\sigma\varphi = \hat{t}\varphi$.

(*2*): If $D\varphi$ is fulfilled, then we get by quasi overlay joinability and Corollary 6.4:

$$t_0\varphi = t_0\varphi[p \leftarrow t_0\varphi/p] \downarrow t_0\varphi[p \leftarrow t_1\varphi/p] = t_1\varphi \;.$$

Proof of Theorem 7.4

For the proof of Sub-claim 3 below, we enrich the signatures by a new sort s_{new} and new constructor symbols $\mathbf{eq}_{\bar{s}}$ for each old sort $\bar{s} \in \mathrm{S}$ with arity $\bar{s}\bar{s} \longrightarrow s_{\mathrm{new}}$ and $\bot$ with arity s_{new}. We take (in addition to R) the following set of new rules (with $x_{\bar{s}} \in \mathrm{V}_{\mathrm{SIG},\bar{s}}$ for $\bar{s} \in \mathrm{S}$):

$$\mathrm{R}' := \{\ \mathbf{eq}_{\bar{s}}\, x_{\bar{s}}\, x_{\bar{s}} = \bot \mid \bar{s} \in \mathrm{S}\ \}$$

Since the sort restrictions do not allow $\Longrightarrow_{\mathrm{R}\cup\mathrm{R}',\mathrm{X},\beta}$ to make any use of terms of the sort s_{new} when rewriting terms of an "old" sort, we get

$$\forall\beta \preceq \omega{+}\omega : \quad \Longrightarrow_{\mathrm{R}\cup\mathrm{R}',\mathrm{X},\beta} \cap (\mathcal{T}(\mathrm{sig},\mathrm{X}){\times}\mathcal{T}(\mathrm{sig},\mathrm{X})) \;=\; \Longrightarrow_{\mathrm{R},\mathrm{X},\beta/\mathrm{sig}/\mathrm{cons}}$$

(the latter being defined over the non-enriched signatures). Therefore (as no new critical peaks occur) the critical peaks keep being quasi overlay joinable. We are going to show confluence of $\Longrightarrow_{\mathrm{R}\cup\mathrm{R}',\mathrm{X}}$, which implies confluence of $\Longrightarrow_{\mathrm{R},\mathrm{X}/\mathrm{sig}/\mathrm{cons}}$. We know that $\Longrightarrow_{\mathrm{R}\cup\mathrm{R}',\mathrm{X}}$ is noetherian on each "old" sort; and by this we also know $\Longrightarrow_{\mathrm{R}\cup\mathrm{R}',\mathrm{X}}$ to be noetherian on s_{new} because $\bot$ is irreducible and terms of the form '$\mathbf{eq}_{\bar{s}}\, u\, v$' allow at most one reduction step via $\Longrightarrow_{\mathrm{R}\cup\mathrm{R}',\mathrm{X}} \setminus \Longrightarrow_{\mathrm{R},\mathrm{X}/\mathrm{sig}'/\mathrm{cons}'}$.

We define $\Longrightarrow_{0} := \Longrightarrow_{\mathrm{R}',\mathrm{X}}$; $\Longrightarrow_{\beta} := \Longrightarrow_{\mathrm{R}\cup\mathrm{R}',\mathrm{X},\beta}$ for any ordinal β with $0 \prec \beta \prec \omega+\omega$ (where $\prec$ is the ordering of ordinal numbers); and $\Longrightarrow\, := \,\Longrightarrow_{\omega+\omega} := \,\Longrightarrow_{\mathrm{R}\cup\mathrm{R}',\mathrm{X}}$.

For $v,u,s,t \in \mathcal{T}$ with $v \stackrel{\circledast}{\Longleftarrow} u$ and $s \stackrel{\circledast}{\Longrightarrow} t$; $\Pi \subseteq \mathcal{O}(u)$ with $\forall p,q \in \Pi : (p \neq q \;\Rightarrow\; p|q)$ and $\forall o \in \Pi : u/o = s$; we say that $P(v,u,s,t,\Pi)$ holds <u>:iff</u> $v \downarrow u[\,o \leftarrow t \mid o \in \Pi]$. Now (by $\Pi := \{\emptyset\}$) it suffices to show that $P(v,u,s,t,\Pi)$ holds for all appropriate v,u,s,t,Π. We will show this by noetherian induction over the lexicographic combination of the following orderings:

1. $(\Longrightarrow \cup \rhd_{\mathrm{ST}})^{\oplus}$ (Cf. Lemma 8.6)
2. $\succ$
3. $\succ$

using the following measure on (v,u,s,t,Π):

1. s
2. the smallest ordinal $\beta \preceq \omega{+}\omega$ for which $v \stackrel{\circledast}{\Longleftarrow}_{\beta} u$
3. the smallest $n \in \mathbb{N}$ for which $v \stackrel{n}{\Longleftarrow}_{\beta} u$ for the β of (2)

For the limit ordinals 0, ω, $\omega{+}\omega$ in the second position of the measure, the induction step is trivial ($\stackrel{\circledast}{\Longleftarrow}_{0} \subseteq \Longleftarrow_{0} \cup \mathrm{id}$; $\stackrel{\circledast}{\Longleftarrow}_{\omega} \subseteq \bigcup_{i\in\mathbb{N}_+} \stackrel{\circledast}{\Longleftarrow}_{i}$; $\stackrel{\circledast}{\Longleftarrow}_{\omega+\omega} \subseteq \bigcup_{i\in\mathbb{N}} \stackrel{\circledast}{\Longleftarrow}_{\omega+i}$). Thus, as we now suppose a smallest (v,u,s,t,Π) with $P(v,u,s,t,\Pi)$ not holding for, the second position of the measure must be a non-limit ordinal $\beta{+}1$.

As $P(v,u,s,t,\Pi)$ holds trivially for $u = v$ or $s = t$ we have some u', s' with $v \stackrel{n}{\Longleftarrow}_{\beta+1} u' \Longleftarrow_{\beta+1} u$ $(n \in \mathbb{N})$ (with $\forall m \in \mathbb{N} : (v \stackrel{m}{\Longleftarrow}_{\beta+1} u \;\Rightarrow\; m > n)$) and $s \Longrightarrow s' \stackrel{\circledast}{\Longrightarrow} t$. Now for a contradiction it is sufficient to show

<u>**Claim:**</u> There is some z with $v \stackrel{\circledast}{\Longrightarrow} z \stackrel{\circledast}{\Longleftarrow} u[\,o \leftarrow s' \mid o \in \Pi]$.

because then we have $z \downarrow u[\,o \leftarrow t \mid o \in \Pi]$ by $P(z, u[\,o \leftarrow s' \mid o \in \Pi], s', t, \Pi)$, which is smaller than (v,u,s,t,Π) in the first position of the measure by $s \Longrightarrow s'$.

If we had $s \Longrightarrow s'$ by some redex below the top of s, then **Claim** would hold by induction hypothesis with $P(\ldots)$ being smaller in the first position using $\rhd_{\mathrm{ST}}$. Thus, there are some $((l_0,r_0),C_0) \in \mathrm{R} \cup \mathrm{R}'$; $\mu_0 \in \mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{X}))$; with $s = l_0\mu_0$; $s' = r_0\mu_0$; $C_0\mu_0$ fulfilled w. r. t. $\Longrightarrow$. Furthermore, we have some $q \in \mathcal{O}(u)$; $((l_1,r_1),C_1) \in \mathrm{R} \cup \mathrm{R}'$; $\mu_1 \in \mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{X}))$; with $u/q = l_1\mu_1$; $u' = u[\,q \leftarrow r_1\mu_1\,]$; $C_1\mu_1$ fulfilled w. r. t. $\Longrightarrow_{\beta}$.

We now distinguish two cases by the relative position of q and Π:

"q not strictly below any $p \in \Pi$": There is no p' with $pp' = q$, $p' \neq \emptyset$, and $p \in \Pi$

Define $\Pi' := \{\ p \mid qp \in \Pi\ \}$.

The critical peak case: There is some $p \in \Pi' \cap \mathcal{O}(l_1)$ with $l_1/p \notin \mathrm{V}$.

Let $\xi := \min \mathrm{Sep}(\mathcal{V}(((l_0,r_0),C_0)), \mathcal{V}(((l_1,r_1),C_1)))$. Let ϱ be given by ($x \in \mathrm{V}$): $x\varrho := \begin{Bmatrix} x\mu_1 & \text{if } x \in \mathcal{V}(((l_1,r_1),C_1)) \\ x\xi^{-1}\mu_0 & \text{otherwise} \end{Bmatrix}$. By $l_0\xi\varrho = l_0\xi\xi^{-1}\mu_0 = s = u/qp = l_1\mu_1/p = (l_1/p)\varrho$ let $\mathrm{Y} := \mathcal{V}(((l_0,r_0),C_0)\xi,((l_1,r_1),C_1))$, $\sigma := \min \mathrm{Mgu}(\langle(l_0\xi, l_1/p)\rangle, \mathrm{Y})$ and $\varphi \in \mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{X}))$ with $\sigma\varphi|_\mathrm{Y} = \varrho|_\mathrm{Y}$. Let $(((t_0,t_1),D),\hat{t}) := (((l_1[p \leftarrow r_0\xi],r_1),C_0\xi\,C_1),l_1)\sigma$. Now we have $(((t_0,t_1),D),\hat{t})\varphi = (((l_1\mu_1[p \leftarrow r_0\mu_0],r_1\mu_1),C_0\mu_0\,C_1\mu_1),l_1\mu_1)$. Therefore $D\varphi$ is fulfilled.

Sub-claim 1: There is some w with $s \stackrel{\oplus}{\Longrightarrow} r_1\mu_1/p \stackrel{\circledast}{\Longrightarrow} w \stackrel{\circledast}{\Longleftarrow} s'$ and $r_1\mu_1 = l_1\mu_1[p \leftarrow r_1\mu_1/p]$.

Proof of Sub-claim 1: If $t_0 = t_1$, then $s \Longrightarrow s' = r_0\mu_0 = t_0\varphi/p = t_1\varphi/p = r_1\mu_1/p$; and $r_1\mu_1 = r_1\mu_1[p \leftarrow r_1\mu_1/p] = t_1\varphi[p \leftarrow r_1\mu_1/p] = t_0\varphi[p \leftarrow r_1\mu_1/p] = l_1\mu_1[p \leftarrow r_1\mu_1/p]$. Otherwise, if $t_0 \neq t_1$, $(((t_0,t_1),D),\hat{t},p)$ is a critical peak and we get $s = u/qp = l_1\mu_1/p = \hat{t}\varphi/p \stackrel{\oplus}{\Longrightarrow} t_1\varphi/p \downarrow (t_0/p)\varphi = r_0\mu_0 = s'$ where $t_1\varphi/p = r_1\mu_1/p$; and $r_1\mu_1 = t_1\varphi = t_0\varphi[p \leftarrow t_1\varphi/p] = l_1\mu_1[p \leftarrow r_1\mu_1/p]$. Q. e. d. (Sub-claim 1)

Sub-claim 2: For $o \in \Pi \setminus \{qp\}$ we have $u'/o = s$.

Proof of Sub-claim 2: If there is some p' with $o = qp'$, then by Sub-claim 1: $u'/o = u'/qp' = r_1\mu_1/p' = l_1\mu_1[p \leftarrow \ldots]/p' = l_1\mu_1/p' = u/qp' = u/o = s$. Otherwise, by $qp \in \Pi$ we know $o|q$, and therefore $u'/o = u/o = s$. Q. e. d. (Sub-claim 2)

By Sub-claim 1 we get $u'[qp \leftarrow w] = u[q \leftarrow r_1\mu_1[p \leftarrow w]] = u[q \leftarrow l_1\mu_1[p \leftarrow w]] = u[qp \leftarrow w]$. Hence, for $\hat{u} := u'[qp \leftarrow w][o \leftarrow s' \mid o \in \Pi \setminus \{qp\}]$ we get $\hat{u} = u[qp \leftarrow w][o \leftarrow s' \mid o \in \Pi \setminus \{qp\}] \stackrel{\circledast}{\Longleftarrow} u[o \leftarrow s' \mid o \in \Pi]$ by $w \stackrel{\circledast}{\Longleftarrow} s'$ (by Sub-claim 1). Thus, for **Claim** we only have to show $v \downarrow \hat{u}$. For $u'' := u'[o \leftarrow s' \mid o \in \Pi \setminus \{qp\}]$ (cf. Sub-claim 2) there is some w' with $v \stackrel{\circledast}{\Longrightarrow} w' \stackrel{\circledast}{\Longleftarrow} u''$ by Sub-claim 2 and $P(v,u',s,s',\Pi \setminus \{qp\})$, which is smaller in the second or third position. Finally, we get $w' \downarrow \hat{u}$ by $u''/qp = u'/qp = r_1\mu_1/p$, Sub-claim 1, and $P(w',u'',r_1\mu_1/p,w,\{qp\})$, which is smaller in the first position by Sub-claim 1. Q. e. d. (The critical peak case)

The variable overlap (if any) case: $\forall p \in \Pi' \cap \mathcal{O}(l_1) : l_1/p \in \mathrm{V}$

Define $\Gamma : \mathrm{V} \to \mathcal{F}(\mathcal{O}[\mathcal{T}])$ by ($x \in \mathrm{V}$): $\Gamma(x) := \{\ p' \mid \exists p : (l_1/p = x \ \wedge\ pp' \in \Pi')\ \}$.

Define μ_1' by ($x \in \mathrm{V}$): $x\mu_1' := x\mu_1[p' \leftarrow s' \mid p' \in \Gamma(x)]$. Define $\hat{u} := u[q \leftarrow r_1\mu_1'][o \leftarrow s' \mid o \in \Pi \setminus (q\Pi')]$ and $\check{u} := u[q \leftarrow l_1\mu_1'][o \leftarrow s' \mid o \in \Pi \setminus (q\Pi')]$.

We are going to show $v \downarrow \hat{u} \Longleftarrow \check{u} \stackrel{\circledast}{\Longleftarrow} u[o \leftarrow s' \mid o \in \Pi]$ for **Claim**. Since for $p' \in \Gamma(x)$ we always have some p with $l_1/p = x$; $x\mu_1/p' = l_1\mu_1/pp' = u/qpp' = s \Longrightarrow s'$; we get $v \downarrow \hat{u}$ by $P(v,u',s,s',(\Pi \setminus (q\Pi')) \cup \{\ qpp' \mid \exists x : (r_1/p = x \ \wedge\ p' \in \Gamma(x))\ \})$, which is smaller in the second or third position. We get $\hat{u} \Longleftarrow \check{u}$ by

Sub-claim 3: $C_1\mu_1'$ is fulfilled.

Finally, $\check{u} \stackrel{\circledast}{\Longleftarrow} u[q \leftarrow l_1\mu_1[o \leftarrow s' \mid o \in \Pi']][o \leftarrow s' \mid o \in \Pi \setminus (q\Pi')] = u[o \leftarrow s' \mid o \in \Pi]$.

Proof of Sub-claim 3: For $(\bar{u}{=}\bar{v})$ in C_1 we have $\bar{u}\mu_1 {\downarrow_\beta} \bar{v}\mu_1$ and hence for the sort $\bar{s}$ of $\bar{u}$: $\perp \stackrel{\oplus}{\Longleftarrow}_\beta (\mathsf{eq}_{\bar{s}}\,\bar{u}\,\bar{v})\mu_1$. We get $\perp \downarrow (\mathsf{eq}_{\bar{s}}\,\bar{u}\,\bar{v})\mu_1'$ by $P(\perp, (\mathsf{eq}_{\bar{s}}\,\bar{u}\,\bar{v})\mu_1, s, s', \{\ pp' \mid \exists x : ((\mathsf{eq}_{\bar{s}}\,\bar{u}\,\bar{v})/p = x \ \wedge\ p' \in \Gamma(x))\ \})$, which is smaller in the second position. Since there are no rules for $\perp$ and only one for $\mathsf{eq}_{\bar{s}}$, this means $\bar{u}\mu_1' \downarrow \bar{v}\mu_1'$. For $(\mathrm{Def}\,\bar{u})$ in C_1 we know the existence of some $\vec{u} \in \mathcal{GT}(\mathrm{cons})$ with $\vec{u} \stackrel{\circledast}{\Longleftarrow}_\beta \bar{u}\mu_1$. We get some $\hat{u}$ with $\vec{u} \stackrel{\circledast}{\Longrightarrow} \hat{u} \stackrel{\circledast}{\Longleftarrow} \bar{u}\mu_1'$ by $P(\vec{u},\bar{u}\mu_1,s,s',\{\ pp' \mid \exists x : (\bar{u}/p = x \ \wedge\ p' \in \Gamma(x))\ \})$, which is smaller in the second position. By Lemma 6.8 we get $\hat{u} \in \mathcal{GT}(\mathrm{cons})$. Finally, for $(\bar{u}{\neq}\bar{v})$ in C_1 we have some $\vec{u},\vec{v} \in \mathcal{GT}(\mathrm{cons})$ with $\bar{u}\mu_1 \stackrel{\circledast}{\Longrightarrow}_\beta \vec{u} \ddagger \vec{v} \stackrel{\circledast}{\Longleftarrow}_\beta \bar{v}\mu_1$. By applying the same procedure as before twice we get $\hat{u},\hat{v} \in \mathcal{GT}(\mathrm{cons})$ with $\bar{u}\mu_1' \stackrel{\circledast}{\Longrightarrow} \hat{u} \stackrel{\circledast}{\Longleftarrow} \vec{u} \ddagger \vec{v} \stackrel{\circledast}{\Longrightarrow} \hat{v} \stackrel{\circledast}{\Longleftarrow} \bar{v}\mu_1'$, i. e. $\bar{u}\mu_1' \stackrel{\circledast}{\Longrightarrow} \hat{u} \ddagger \hat{v} \stackrel{\circledast}{\Longleftarrow} \bar{v}\mu_1'$.

Q. e. d. (Sub-claim 3; The variable overlap (if any) case; "q not strictly below any $p \in \Pi$")

"q strictly below $p \in \Pi$": There is some p' with $pp' = q$, $p' \neq \emptyset$, and $p \in \Pi$

Sub-claim 4: For $o \in \Pi \setminus \{p\}$ we have $u'/o = s = l_0\mu_0$.

Proof of Sub-claim 4: Since $o|p$, we have $u'/o = u[\,pp' \leftarrow \ldots]/o = u/o = s = l_0\mu_0$.

Q. e. d. (Sub-claim 4)

The (second) critical peak case: $p' \in \mathcal{O}(l_0) \ \wedge \ l_0/p' \notin \mathrm{V}$

Let $\xi := \min \mathrm{Sep}(\mathcal{V}(((l_1, r_1), C_1)), \mathcal{V}(((l_0, r_0), C_0)))$. Let ϱ be given by $(x \in \mathrm{V})$: $x\varrho := \begin{Bmatrix} x\mu_0 & \text{if } x \in \mathcal{V}(((l_0, r_0), C_0)) \\ x\xi^{-1}\mu_1 & \text{otherwise} \end{Bmatrix}$. By $l_1\xi\varrho = l_1\xi\xi^{-1}\mu_1 = u/q = u/pp' = l_0\mu_0/p' = (l_0/p')\varrho$ let $\mathrm{Y} := \mathcal{V}(((l_1, r_1), C_1)\xi, ((l_0, r_0), C_0))$, $\sigma := \min \mathrm{Mgu}(\langle(l_1\xi, l_0/p')\rangle, \mathrm{Y})$ and $\varphi \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}(\mathrm{X}))$ with $\sigma\varphi|_{\mathrm{Y}} = \varrho|_{\mathrm{Y}}$. Let $(((t_0, t_1), D), \hat{t}) := (((l_0[\,p' \leftarrow r_1\xi\,], r_0), C_1\xi\, C_0), l_0)\sigma$. Now we have $(((t_0, t_1), D), \hat{t})\varphi = (((l_0\mu_0[\,p' \leftarrow r_1\mu_1\,], r_0\mu_0), C_1\mu_1\, C_0\mu_0), l_0\mu_0)$. Therefore $D\varphi$ is fulfilled.

Sub-claim 5: There is some w with $s \Longrightarrow s[\,p' \leftarrow r_1\mu_1\,] \overset{\circledast}{\Longrightarrow} w \overset{\circledast}{\Longleftarrow} s'$.

Proof of Sub-claim 5: Since $s/p' = u/pp' = u/q = l_1\mu_1$ we have $s \Longrightarrow s[\,p' \leftarrow r_1\mu_1\,]$. Because $s[\,p' \leftarrow r_1\mu_1\,] = l_0\mu_0[\,p' \leftarrow r_1\mu_1\,] = t_0\varphi$ and $t_1\varphi = r_0\mu_0 = s'$ we now only have to show $t_0\varphi \downarrow t_1\varphi$. If $t_0 = t_1$, this is trivial. Otherwise, if $t_0 \neq t_1$, $(((t_0, t_1), D), \hat{t}, p')$ is a critical peak and we get $t_0\varphi \downarrow t_1\varphi$ by Lemma 7.3. Q. e. d. (Sub-claim 5)

For $\hat{u} := u[\,p \leftarrow w\,][\,o \leftarrow s' \mid o \in \Pi \setminus \{p\}]$ we get $\hat{u} \overset{\circledast}{\Longleftarrow} u[\,o \leftarrow s' \mid o \in \Pi]$ by $w \overset{\circledast}{\Longleftarrow} s'$ (by Sub-claim 5). Thus, for **Claim** we only have to show $v \downarrow \hat{u}$. For $u'' := u'[\,o \leftarrow s' \mid o \in \Pi \setminus \{p\}]$ (Cf. Sub-claim 4) there is some w' with $v \overset{\circledast}{\Longrightarrow} w' \overset{\circledast}{\Longleftarrow} u''$ by Sub-claim 4 and $P(v, u', s, s', \Pi \setminus \{p\})$, which is smaller in the second or third position. Finally, we get $w' \downarrow \hat{u}$ by $u''/p = u'/p = u[\,pp' \leftarrow r_1\mu_1\,]/p = u/p[\,p' \leftarrow r_1\mu_1\,] = s[\,p' \leftarrow r_1\mu_1\,]$, Sub-claim 5, and $P(w', u'', s[\,p' \leftarrow r_1\mu_1\,], w, \{p\})$, which is smaller in the first position by Sub-claim 5.

Q. e. d. (The (second) critical peak case)

The (second) variable overlap case: There are $\check{p}, \hat{p}, x$ with $p' = \check{p}\hat{p}$ and $l_0/\check{p} = x \in \mathrm{V}$

Define μ'_0 by $(y \in \mathrm{V})$: $y\mu'_0 := \begin{Bmatrix} x\mu_0[\,\hat{p} \leftarrow r_1\mu_1\,] & \text{if } y = x \\ y\mu_0 & \text{otherwise} \end{Bmatrix}$. Define $\hat{u} := u[\,o \leftarrow r_0\mu'_0 \mid o \in \Pi]$ and $\check{u} := u[\,o \leftarrow l_0\mu'_0 \mid o \in \Pi]$. Since $x\mu_0/\hat{p} = l_0\mu_0/\check{p}\hat{p} = s/p' = u/pp' = u/q = l_1\mu_1 \Longrightarrow r_1\mu_1$ and $u' = u[\,p\check{p}\hat{p} \leftarrow r_1\mu_1\,] = u[\,p \leftarrow l_0\mu_0[\,\check{p} \leftarrow x\mu'_0\,]\,][\,o \leftarrow l_0\mu_0 \mid o \in \Pi \setminus \{p\}]$ (by Sub-claim 4), we get some w with $v \overset{\circledast}{\Longrightarrow} w \overset{\circledast}{\Longleftarrow} \check{u}$ by $P(v, u', l_1\mu_1, r_1\mu_1, \{\, o\check{o}\hat{p} \mid o \in \Pi \wedge l_0/\check{o} = x \wedge o\check{o} \neq p\check{p} \,\})$, which is smaller in the first position by $s \rhd_{\mathrm{ST}} s/p' = u/pp' = u/q = l_1\mu_1$. Now by $\hat{u} \overset{\circledast}{\Longleftarrow} u[\,o \leftarrow r_0\mu_0 \mid o \in \Pi] = u[\,o \leftarrow s' \mid o \in \Pi]$ for **Claim** we only have to show $w \downarrow \hat{u}$. But this is given by Sub-claim 6 below and $P(w, \check{u}, l_0\mu'_0, r_0\mu'_0, \Pi)$, which is smaller in the first position of the measure by $s = l_0\mu_0 \overset{\oplus}{\Longrightarrow} l_0\mu'_0$.

Sub-claim 6: $C_0\mu'_0$ is fulfilled.

Proof of Sub-claim 6: For $(\bar{u}{=}\bar{v})$ in C_0 we have some w with $\bar{u}\mu_0 \overset{\circledast}{\Longrightarrow} w \overset{\circledast}{\Longleftarrow} \bar{v}\mu_0$. We get some w' with $w \overset{\circledast}{\Longrightarrow} w' \overset{\circledast}{\Longleftarrow} \bar{u}\mu'_0$ by $P(w, \bar{u}\mu_0, l_1\mu_1, r_1\mu_1, \{\, o\hat{p} \mid \bar{u}/o = x \,\})$, and then $w' \downarrow \bar{v}\mu'_0$ by $P(w', \bar{v}\mu_0, l_1\mu_1, r_1\mu_1, \{\, o\hat{p} \mid \bar{v}/o = x \,\})$, which are smaller in the first position of the measure by $s \rhd_{\mathrm{ST}} l_1\mu_1$ (shown above). For $(\mathrm{Def}\,\bar{u})$ in C_0 we know the existence of some $\vec{\bar{u}} \in \mathcal{GT}(\mathrm{cons})$ with $\vec{\bar{u}} \overset{\circledast}{\Longleftarrow} \bar{u}\mu_0$. We get some $\hat{\bar{u}}$ with $\vec{\bar{u}} \overset{\circledast}{\Longrightarrow} \hat{\bar{u}} \overset{\circledast}{\Longleftarrow} \bar{u}\mu'_0$ by $P(\vec{\bar{u}}, \bar{u}\mu_0, l_1\mu_1, r_1\mu_1, \{\, o\hat{p} \mid \bar{u}/o = x \,\})$, which is smaller in the first position (as before). By Lemma 6.8 we get $\hat{\bar{u}} \in \mathcal{GT}(\mathrm{cons})$. Finally, for $(\bar{u}{\neq}\bar{v})$ in C_0 we have some $\vec{\bar{u}}, \vec{\bar{v}} \in \mathcal{GT}(\mathrm{cons})$ with $\bar{u}\mu_0 \overset{\circledast}{\Longrightarrow} \vec{\bar{u}} \ddagger \vec{\bar{v}} \overset{\circledast}{\Longleftarrow} \bar{v}\mu_0$. By applying the same procedure as before twice we get $\hat{\bar{u}}, \hat{\bar{v}} \in \mathcal{GT}(\mathrm{cons})$ with $\bar{u}\mu'_0 \overset{\circledast}{\Longrightarrow} \hat{\bar{u}} \overset{\circledast}{\Longleftarrow} \vec{\bar{u}} \ddagger \vec{\bar{v}} \overset{\circledast}{\Longrightarrow} \hat{\bar{v}} \overset{\circledast}{\Longleftarrow} \bar{v}\mu'_0$, i. e. $\bar{u}\mu'_0 \overset{\circledast}{\Longrightarrow} \hat{\bar{u}} \ddagger \hat{\bar{v}} \overset{\circledast}{\Longleftarrow} \bar{v}\mu'_0$.

Q. e. d. (Sub-claim 6)

Q. e. d. (The (second) variable overlap case)

Q. e. d. ("q strictly below $p \in \Pi$")

Q. e. d. **(Proof of Theorem 7.4)**

Proof of Theorem 7.6

Claim: For $\beta \preceq \omega{+}\omega$ and $s \overset{\circledast}{\Longrightarrow}_{R,X,\beta} t$ we have $\mathcal{A}_\kappa(s) = \mathcal{A}_\kappa(t)$.

Proof of Claim: By induction on β. By induction on the number of derivation steps, it suffices to do the proof for $\Longrightarrow_{R,X,\beta}$ instead of $\overset{\circledast}{\Longrightarrow}_{R,X,\beta}$. If β is one of the limit ordinals 0, ω, $\omega{+}\omega$, the induction step is trivial. If β is a non-limit ordinal $\gamma{+}1$, the induction step is as follows: For $s \Longrightarrow_{R,X,\gamma+1} t$, either $s \Longrightarrow_{R,X,\omega} t$ and $\omega \preceq \gamma$, and then (by induction hypothesis) $\mathcal{A}_\kappa(s) = \mathcal{A}_\kappa(t)$; or there is a substitution $\sigma \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}(\mathrm{X}))$, a rule $((l,r),C) \in \mathrm{R}$, and a $p \in \mathcal{O}(t)$ with $s/p = l\sigma$, $t = s[p \leftarrow r\sigma]$, and $C\sigma$ fulfilled w. r. t. $\Longrightarrow_{R,X,\gamma}$. Since $\mathcal{A}$ is a sig/cons-model of R, we only have to show that $C\sigma$ is true w. r. t. $\mathcal{A}_\kappa$. For $(u{=}v)$ in $C\sigma$ we have $u \downarrow_{R,X,\gamma} v$ and hence by induction hypothesis we have $\mathcal{A}_\kappa(u) = \mathcal{A}_\kappa(v)$. For $\bar{u} \in \mathrm{S}$, $u \in \mathcal{T}(\mathrm{sig},\mathrm{X})_{\bar{u}}$, $(\mathrm{Def}\,u)$ in $C\sigma$ there is some $\hat{u} \in \mathcal{GT}_{\mathrm{CONS},\bar{u}}$ with $u \overset{\circledast}{\Longrightarrow}_{R,X,\gamma} \hat{u}$ and hence by induction hypothesis we have $\mathcal{A}_\kappa(u) = \mathcal{A}_\kappa(\hat{u}) \in \mathcal{A}(\mathrm{CONS},\bar{u})$. For $(u{\neq}v)$ in $C\sigma$ we know $\omega \preceq \gamma$ and there are $\hat{u},\hat{v} \in \mathcal{GT}(\mathrm{cons})$ with $u \overset{\circledast}{\Longrightarrow}_{R,X,\gamma} \hat{u} \not\updownarrow_{R,X,\gamma} \hat{v} \overset{\circledast}{\Longleftarrow}_{R,X,\gamma} v$ and w. l. o. g. (since $\Longrightarrow_{R,X,\omega}$ is noetherian and by Lemma 6.8) $\hat{u},\hat{v} \notin \mathrm{dom}(\Longrightarrow_{R,X})$; thus, (since $\hat{u}$ and $\hat{v}$ are of the same sort and unequal) we have by induction hypothesis: $\mathcal{A}_\kappa(u) = \mathcal{A}_\kappa(\hat{u}) \neq \mathcal{A}_\kappa(\hat{v}) = \mathcal{A}_\kappa(v)$.
Q. e. d. (Claim)

Now we show confluence for part (1) of the theorem. Suppose $u \overset{\oplus}{\Longleftarrow}_{R,X} s \overset{\oplus}{\Longrightarrow}_{R,X} v$. Since $\Longrightarrow_{R,X}$ is noetherian, there are $\hat{u},\hat{v} \in \mathcal{T}(\mathrm{sig},\mathrm{X}) \setminus \mathrm{dom}(\Longrightarrow_{R,X})$ with $u \overset{\circledast}{\Longrightarrow}_{R,X} \hat{u}$ and $v \overset{\circledast}{\Longrightarrow}_{R,X} \hat{v}$. By Claim we have $\mathcal{A}_\kappa(\hat{u}) = \mathcal{A}_\kappa(u) = \mathcal{A}_\kappa(s) = \mathcal{A}_\kappa(v) = \mathcal{A}_\kappa(\hat{v})$ and therefore $\hat{u} = \hat{v}$.
Finally for the proof of part (2) of the theorem, in the above we can additionally assume $s \in D_{\mathrm{X}}$ and thus the existence of some $t \in \mathcal{T}(\mathrm{cons}, \mathrm{V}_{\mathrm{CONS}} \cap \mathrm{X}) \setminus \mathrm{dom}(\Longrightarrow_{R,X})$ (since $\Longrightarrow_{R,X,\omega}$ is noetherian and by Lemma 6.7) with $s \overset{\circledast}{\Longleftrightarrow}_{R,X} t$. Then by Claim we have $\mathcal{A}_\kappa(\hat{u}) = \mathcal{A}_\kappa(u) = \mathcal{A}_\kappa(s) = \mathcal{A}_\kappa(t)$ and $\mathcal{A}_\kappa(\hat{v}) = \mathcal{A}_\kappa(v) = \mathcal{A}_\kappa(s) = \mathcal{A}_\kappa(t)$, i. e. $\hat{u} = t = \hat{v}$.

Proof of Lemma 8.1 and Lemma 8.2

It is standard to encode a universal deterministic Turing machine with a finite set of left-linear, non-overlapping rules. This can be done in the following way, where stop, left, right, nil, 0, ⊥ are constant symbols, s is a unary function symbol, cons and nth are binary, cmd, state are ternary, and T is sexary; and " T l a r c s p " encodes the Turing machine with

	meaning	intended range
l	being the tape to the left of the head	nil or cons s^n 0 list-of-integers
a	being the symbol under the head	s^n 0
r	being the tape to the right of the head	nil or cons s^n 0 list-of-integers
c	being the next command to be executed	stop or left or right or s^n 0
s	being the next state to be in	s^n 0
p	being the program	cons table-of-commands table-of-states

The rules are:

T	l	a	r	stop	$s\ p =$	⊥							
T	nil	a	r	left	$s\ p =$	T	nil	0	cons $a\ r$	cmd 0 $s\ p$	state 0 $s\ p$	p
T	cons $b\ l$	a	r	left	$s\ p =$	T	l	b	cons $a\ r$	cmd $b\ s\ p$	state $b\ s\ p$	p
T	l	a	nil	right	$s\ p =$	T	cons $a\ l$	0	nil	cmd 0 $s\ p$	state 0 $s\ p$	p
T	l	a	cons $b\ r$	right	$s\ p =$	T	cons $a\ l$	b	r	cmd $b\ s\ p$	state $b\ s\ p$	p
T	l	a	r	0	$s\ p =$	T	l	0	r	cmd 0 $s\ p$	state 0 $s\ p$	p
T	l	a	r	sx	$s\ p =$	T	l	sx	r	cmd s$x\ s\ p$	state s$x\ s\ p$	p

with the following auxiliary functions:

$$
\begin{aligned}
\mathsf{nth}\ \ 0\ \ \mathsf{cons}\, a\, l &= a \\
\mathsf{nth}\ \ \mathsf{s}\, x\ \ \mathsf{cons}\, a\, l &= \mathsf{nth}\ \ x\ \ l \\
\mathsf{cmd}\ \ a\ \ s\ \ \mathsf{cons}\ \textit{table-of-commands}\ \textit{table-of-states} &= \mathsf{nth}\ \ a\ \ \mathsf{nth}\ \ s\ \ \textit{table-of-commands} \\
\mathsf{state}\ \ a\ \ s\ \ \mathsf{cons}\ \textit{table-of-commands}\ \textit{table-of-states} &= \mathsf{nth}\ \ a\ \ \mathsf{nth}\ \ s\ \ \textit{table-of-states}
\end{aligned}
$$

We use $\bot$ instead of a reasonable output because we are interested in the halting problem only. We assume all function symbols so far to be constructor symbols.

From this system we are now going to construct our positive-conditional rule system by exchanging the recursive T-rules of the form " T l a r c s p = T l′ a′ r′ c′ s′ p′ " for rules of the form " T l a r c s p $=\ \bot \longleftarrow$ T l′ a′ r′ c′ s′ p′ $=\ \bot$ ".

Now the above Turing machine halts iff the ground term " T l a r c s p " is reducible. Therefore (the halting problem being not co-semi-decidable) the reducibility of ground terms cannot be co-semi-decidable.

For Lemma 8.2 we now add the following new rule:

$$\mathsf{foreverp}\ l\ a\ r\ c\ s\ p = \mathsf{true} \longleftarrow \mathsf{T}\ l\ a\ r\ c\ s\ p \neq \bot,\ \ \mathrm{Def}\,\mathsf{T}\ l\ a\ r\ c\ s\ p,\ \ \mathrm{Def}\,\bot$$

Now " foreverp l a r c s p " is reducible iff the above Turing machine does not halt. Therefore the reducibility of ground terms cannot be semi-decidable.

Proof of Theorem 8.3

Proof of (*1.*): The function g that firstly tests whether its single argument s is in the enumerable set $\mathcal{T}(\mathrm{sig},\mathrm{X})$, secondly tries to compute $f(s)$, and thirdly (if $s \in \mathcal{T}(\mathrm{sig},\mathrm{X})$ and $f(s)$ is defined) is defined iff the test $f(s) = s$ succeeds, is defined exactly on the irreducible terms from $\mathcal{T}(\mathrm{sig},\mathrm{X})$.

Proof of (*2.*): Claim 1: Let $\beta \preceq \omega{+}\omega$. For all $s \in \mathcal{T}(\mathrm{sig},\mathrm{X})$ the fol. sets are enumerable:

$$Y_\beta(s) := \{\ t \mid s \Longrightarrow_{\mathrm{R,X},\beta} t\ \}$$

$$Z_\beta(s) := \{\ t \mid s \overset{\circledast}{\Longrightarrow}_{\mathrm{R,X},\beta} t\ \}$$

If Claim 1 holds, we can enumerate $Z_{\omega+\omega}(s)$ and simultaneously the irreducible terms from $\mathcal{T}(\mathrm{sig},\mathrm{X})$ until one term t occurs in both enumerations and we can return $f(s) := t$.

Proof of Claim 1: By induction on β. It suffices to show that $Y_\beta(s)$ is enumerable for all $s \in \mathcal{T}(\mathrm{sig},\mathrm{X})$. The induction step for the limit ordinals 0, ω, $\omega{+}\omega$ is trivial ($Y_0(s) = \emptyset$; $Y_\omega(s) = \bigcup_{i \prec \omega} Y_i(s)$; $Y_{\omega+\omega}(s) = \bigcup_{i \prec \omega} Y_{\omega+i}(s)$). The induction step for the non-limit ordinals $\beta{+}1$ can be done the following way: For all rules $((l,r),C) \in \mathrm{R}$ and all $p \in \mathcal{O}(s)$ and all σ in the enumerable set $\mathcal{SUB}(\mathcal{V}(((l,r),C)),\mathcal{T}(\mathrm{X}))$ with $s/p = l\sigma$ we test in an enumerative fashion whether $C\sigma$ is fulfilled w. r. t. $\Longrightarrow_{\mathrm{R,X},\beta}$ and enumerate $s[p \leftarrow r\sigma]$ if the test succeeds. For $\omega \preceq \beta$ we merge this enumeration with that of $Y_\omega(s)$. The test of "$C\sigma$ fulfilled w. r. t. $\Longrightarrow_{\mathrm{R,X},\beta}$" can be semi-decided the following way: For $(u{=}v)$ in $C\sigma$ we test the enumerable set $Z_\beta(u) \cap Z_\beta(v)$ for non-emptiness. For $(\mathrm{Def}\, u)$ in $C\sigma$ we test the enumerable set $Z_\beta(u) \cap \mathcal{GT}(\mathrm{cons})$ for non-emptiness. For $(u{\neq}v)$ in $C\sigma$ we test for the existence of $\hat{u} \in A(u)$ and $\hat{v} \in A(v)$ with $\hat{u} \neq \hat{v}$ (syntactically) for the enumerable sets $A(w) := Z_\beta(w) \cap (\mathcal{GT}(\mathrm{cons}) \setminus \mathrm{dom}(\Longrightarrow_{\mathrm{R,X}}))$. This last test succeeds only if $\exists \hat{u}, \hat{v} \in \mathcal{GT}(\mathrm{cons}) : u \overset{\circledast}{\Longrightarrow}_{\mathrm{R,X},\beta} \hat{u} \updownarrow_{\mathrm{R,X},\beta} \hat{v} \overset{\circledast}{\Longleftarrow}_{\mathrm{R,X},\beta} v$ holds. It also succeeds if this property holds because of $\forall s \in \mathcal{GT}(\mathrm{cons}) : \exists t : s \overset{\circledast}{\Longrightarrow}_{\mathrm{R,X}} t \notin \mathrm{dom}(\Longrightarrow_{\mathrm{R,X}})$, Lemma 6.8, and $\omega \preceq \beta$.

Proof of Lemma 8.5

There exists some $\tau \in \mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{X}))$. It suffices to show the fol. claims for all $t, t' \in \mathcal{T}$:

Claim 1: For $t \Longrightarrow_{\mathrm{R,V}} t'$ we have $t\tau \Longrightarrow_{\mathrm{R,X}} t'\tau$.

Claim 2: For $q \in \mathcal{O}(t)$ and $t/q \twoheadrightarrow_{\mathrm{R,V}} t'$ we have $t\tau \twoheadrightarrow_{\mathrm{R,X}} t'\tau$.

Proof of Claim 1: By Corollary 6.5 we get $t\tau \Longrightarrow_{R,V} t'\tau$ and then by Lemma 6.12 $t\tau \Longrightarrow_{R,X} t'\tau$.

Proof of Claim 2: There are $((l,r),C) \in R$; $\sigma \in \mathcal{SUB}(V,\mathcal{T})$; $p \in \mathcal{O}(t/q)$; $u \in \mathcal{TERMS}(C)$ with $t/qp = l\sigma$; $t' = u\sigma$; $C\sigma$ fulfilled w. r. t. $\Longrightarrow_{R,V}$. By Corollary 6.5, $C\sigma\tau$ is fulfilled w. r. t. $\Longrightarrow_{R,V}$. By Lemma 6.12, $C\sigma\tau$ is fulfilled w. r. t. $\Longrightarrow_{R,X}$. Thus; since $t\tau \in \mathcal{T}(\mathrm{sig}, X)$; $\sigma\tau \in \mathcal{SUB}(V, \mathcal{T}(X))$; $qp \in \mathcal{O}(t\tau)$; $t\tau/qp = l\sigma\tau$; $t'\tau = u\sigma\tau$; we get $t\tau \twoheadrightarrow_{R,X} t'\tau$.

Proof of Lemma 8.6

1.: Suppose that $\succ$ is not noetherian. As $\rhd_{ST}$ and $\rightrightarrows$ are noetherian, there must be $r, s : \mathbb{N} \to \mathcal{T}$ with $\forall i \in \mathbb{N} : r_i \rhd_{ST} s_i \rightrightarrows^{\oplus} r_{i+1}$. Then there is a $p : \mathbb{N} \to \mathbb{N}_+^+$ with $\forall i \in \mathbb{N} : r_i/p_i = s_i$. Define $t_n := r_0[p_0 \leftarrow r_1[p_1 \leftarrow r_2 \ldots [p_{n-1} \leftarrow r_n] \ldots]]$. Because of $r_i/p_i = s_i \rightrightarrows^{\oplus} r_{i+1}$ we get $t_n \in \mathcal{T}$ (as $\rightrightarrows^{\oplus}$ sort-invariant) and $t_n \rightrightarrows^{\oplus} t_{n+1}$ (as $\rightrightarrows^{\oplus}$ V-monotonic). This contradicts $\rightrightarrows$ being noetherian.
If $\rightrightarrows$ is V-stable, additionally, then $\succ$ is V-stable too, because $\rhd_{ST}$ is.
Here is an example for $\succ$ not sort-invariant and not $\emptyset$-monotonic: Let A, B be two different sorts. Let $\alpha(\mathsf{a}) = A$, $\alpha(\mathsf{f}) = A \longrightarrow B$, $\alpha(\mathsf{g}) = A \longrightarrow A$. Define $\rightrightarrows := \emptyset$. Then we have $\succ = \rhd_{ST}$ and therefrom: $\mathsf{f\,a} \succ \mathsf{a}$ (hence not sort-invariant); and $\mathsf{g\,a} \succ \mathsf{a}$ but $\mathsf{f\,g\,a} \not\succ \mathsf{f\,a}$ (hence not $\emptyset$-monotonic).
2.: Take the signature from the example in the proof of (1). $\rightrightarrows := \{(\mathsf{a}, \mathsf{f\,a})\}$ is a V-monotonic (indeed!), well-founded ordering on $\mathcal{T}$ that is not sort-invariant. Now $\succ$ is not irreflexive: $\mathsf{a} \rightrightarrows \mathsf{f\,a} \rhd_{ST} \mathsf{a}$.
If one changes $\alpha(\mathsf{f})$ to be $\alpha(\mathsf{f}) = A \longrightarrow A$, then $\rightrightarrows$ is a sort-invariant, well-founded ordering on $\mathcal{T}$ that is not V-monotonic.
3.: For $t \rhd_{ST} t' \rightrightarrows t''$ there is a $p \in \mathcal{O}(t)$; $p \neq \emptyset$ with $t' = t/p$. By sort-invariance and V-monotonicity of $\rightrightarrows$ we get $t = t[p \leftarrow t'] \rightrightarrows t[p \leftarrow t''] \rhd_{ST} t''$.
4.: W. r. t. the noetherian word rewriting system $\{(\rhd_{ST}\rightrightarrows,\ \rightrightarrows\rhd_{ST})\}$, generated from (*3*), words from $\{\rightrightarrows, \rhd_{ST}\}^+$ have normal forms only in $\{\rhd_{ST}\}^+ \cup (\{\rightrightarrows\}^+ \circ \{\rhd_{ST}\}^*)$.

Proof of Lemma 8.7

Suppose $\rhd$ is not noetherian. Then there is a $t : \mathbb{N} \to \mathcal{T}$ with

$$\forall i \in \mathbb{N} : t_i(\rightrightarrows \cup \hookrightarrow \cup \rhd_{ST})\, t_{i+1} .$$

Define $k : \mathbb{N} \to \mathbb{N}$ by:

$$k_0 = 0$$
$$k_{i+1} = 1 + \min\{ j \mid j \geq k_i \wedge t_j \hookrightarrow t_{j+1}\}$$

The above minimum always exists because $(\rightrightarrows \cup \rhd_{ST})$ is noetherian. We have $t_{k_i}((\rightrightarrows \cup \rhd_{ST})^{\circledast} \circ \hookrightarrow)\, t_{k_{i+1}}$. By $\rhd_{ST} \circ \rightrightarrows \subseteq \rightrightarrows \circ \rhd_{ST}$ we get $t_{k_i}(\rightrightarrows^{\circledast} \circ \rhd_{ST}{}^{\circledast} \circ \hookrightarrow)\, t_{k_{i+1}}$, which means $t_{k_i}(\rightrightarrows \cup (\trianglerighteq_{ST} \circ \hookrightarrow))^{\oplus} t_{k_{i+1}}$, which contradicts the assertion that $(\rightrightarrows \cup (\trianglerighteq_{ST} \circ \hookrightarrow))$ is noetherian.
An alternative proof (using the Axiom of Choice) can be done the following way: $(\rightrightarrows \cup (\trianglerighteq_{ST} \circ \hookrightarrow))$ *quasi-commutes* over $(\rightrightarrows \cup \rhd_{ST})$ in the sense of Bachmair&Dershowitz[2]. Thus, $(\rightrightarrows \cup \hookrightarrow \cup \rhd_{ST})$ must be noetherian by Theorem 1 of Bachmair&Dershowitz[2].

Proof of Lemma 8.15

$\Longrightarrow_{R,Y} \subseteq >$ is trivial by induction on the construction of $\Longrightarrow_{R,Y}$ using Lemma 6.10. $\Longrightarrow_{R,Y} \cup \twoheadrightarrow_{R,Y} \cup \rhd_{ST} \subseteq \rhd$ is trivial. R is X-compatible with T by Lemma 6.12. By Lemma 8.5 we know that $(\Longrightarrow_{R,V} \cup (\trianglerighteq_{ST} \circ \twoheadrightarrow_{R,V}))$ is noetherian. By Lemma 8.6 and Lemma 8.7 we know that $(\overset{\oplus}{\Longrightarrow}_{R,V}, (\Longrightarrow_{R,V} \cup \twoheadrightarrow_{R,V} \cup \rhd_{ST})^{\oplus})$ is a termination-pair over sig/V, with which R is trivially V-compatible.

Proof of Theorem 8.17

(1) ⇒ *(2)*: By Lemma 7.2. Q. e. d. ((*1*) ⇒ (*2*))

(*2*) ⇒ (*1*): First notice that the usual modularisation of the proof for the unconditional analogue of the theorem (by showing first that local confluence is guaranteed except for the cases that are matched by critical peaks (the so-called "critical pair lemma")) is not possible here because we need the confluence-property to hold for the condition terms even for the cases that are not matched by critical peaks. Now to the proof: Let s be minimal in $\rhd$ such that $\Longrightarrow$ is not confluent below s. Because of $\Longrightarrow\ \subseteq\ \rhd$ (by Lemma 8.15), $\Longrightarrow$ is not even locally confluent below s. Let $p, q \in \mathcal{O}(s)$; $t_0 \Longleftarrow_p s \Longrightarrow_q t_1$; $t_0 \not\downarrow t_1$. Now as one of p, q must be a prefix of the other, w. l. o. g. say that q is a prefix of p. As $s \trianglerighteq s/q$, by the minimality of s we have $q = \emptyset$. Now for $k < 2$ there must be $((l_k, r_k), C_k) \in \mathrm{R}$; $\mu_k \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}(\mathrm{X}))$; with $C_k\mu_k$ fulfilled; $s = l_1\mu_1$; $s/p = l_0\mu_0$; $t_0 = l_1\mu_1[p \leftarrow r_0\mu_0]$; $t_1 = r_1\mu_1$.

The inductive case: $p = q_0q_1$; $l_1/q_0 = x \in \mathrm{V}$: By $x\mu_1/q_1 = l_1\mu_1/q_0q_1 = s/p = l_0\mu_0$ and Lemma 6.7 (in case of $x \in \mathrm{V_{CONS}}$), we can define $\nu \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}(\mathrm{X}))$ by ($y \in \mathrm{V}$): $y\nu := \left\{ \begin{array}{ll} x\mu_1[q_1 \leftarrow r_0\mu_0] & \text{if } y = x \\ y\mu_1 & \text{otherwise} \end{array} \right\}$ and get $y\mu_1 \overset{\leq 1}{\Longrightarrow} y\nu$ for $y \in \mathrm{V}$. By Corollary 6.4:

$t_0 = l_1\mu_1[q_0q_1 \leftarrow r_0\mu_0] = l_1[q_0 \leftarrow x\nu][q' \leftarrow y\mu_1 \mid l_1/q' = y \in \mathrm{V} \wedge q' \neq q_0]$

$\overset{\circledast}{\Longrightarrow} l_1[q' \leftarrow y\nu \mid l_1/q' = y \in \mathrm{V}] = l_1\nu$;

$t_1 = r_1\mu_1 \overset{\circledast}{\Longrightarrow} r_1\nu$. It suffices to show $l_1\nu \Longrightarrow r_1\nu$, which follows from:

Claim: $C_1\nu$ is fulfilled.

Proof of Claim: For each L in C_1 we have to show that $L\nu$ is fulfilled. By our compatibility-property we get $\forall u \in \mathcal{TERMS}(L) : l_1\mu_1 \rhd u\mu_1$.

$L = (u{=}v)$: We know $u\nu \overset{\circledast}{\Longleftarrow} u\mu_1 \downarrow v\mu_1 \overset{\circledast}{\Longrightarrow} v\nu$. As $s = l_1\mu_1 \rhd u\mu_1$ we have $u\nu \downarrow v\mu_1 \overset{\circledast}{\Longrightarrow} v\nu$. As $s = l_1\mu_1 \rhd v\mu_1$ we have $u\nu \downarrow v\nu$.

$L = (\mathrm{Def}\, u)$: We know the existence of $\hat{u} \in \mathcal{GT}(\mathrm{cons})$ with $u\nu \overset{\circledast}{\Longleftarrow} u\mu_1 \overset{\circledast}{\Longrightarrow} \hat{u}$. By Lemma 6.8 and Lemma 8.15 we can additionally assume $\hat{u}$ to be irreducible. By $s = l_1\mu_1 \rhd u\mu_1$ we get $u\nu \overset{\circledast}{\Longrightarrow} \hat{u}$.

$L = (u{\neq}v)$: We know the existence of $\hat{u}, \hat{v} \in \mathcal{GT}(\mathrm{cons})$ with $u\nu \overset{\circledast}{\Longleftarrow} u\mu_1 \overset{\circledast}{\Longrightarrow} \hat{u} \not\downarrow \hat{v} \overset{\circledast}{\Longleftarrow} v\mu_1 \overset{\circledast}{\Longrightarrow} v\nu$. Just like above we can additionally assume $\hat{u}, \hat{v}$ to be irreducible. By $s = l_1\mu_1 \rhd u\mu_1, v\mu_1$ we get $u\nu \downarrow \hat{u}$; $\hat{v} \downarrow v\nu$; and hence $u\nu \overset{\circledast}{\Longrightarrow} \hat{u} \not\downarrow \hat{v} \overset{\circledast}{\Longleftarrow} v\nu$.

Q. e. d. (Claim; The inductive case)

The critical peak case: $p \in \mathcal{O}(l_1)$; $l_1/p \notin \mathrm{V}$: Let $\xi := \min \mathrm{Sep}(\mathcal{V}(((l_0, r_0), C_0)), \mathcal{V}(((l_1, r_1), C_1)))$ and $\mathrm{Y} := \mathcal{V}(((l_0, r_0), C_0)\xi, ((l_1, r_1), C_1))$. Let ϱ be given by $x\varrho = \left\{ \begin{array}{ll} x\mu_1 & \text{if } x \in \mathcal{V}(((l_1, r_1), C_1)) \\ x\xi^{-1}\mu_0 & \text{otherwise} \end{array} \right\}$ ($x \in \mathrm{V}$).

By $l_0\xi\varrho = l_0\xi\xi^{-1}\mu_0 = s/p = l_1\mu_1/p = l_1\varrho/p = (l_1/p)\varrho$ let $\sigma := \min \mathrm{Mgu}(\langle(l_0\xi, l_1/p)\rangle, \mathrm{Y})$ and $\tau \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}(\mathrm{X}))$ with $\sigma\tau|_{\mathrm{Y}} = \varrho|_{\mathrm{Y}}$.

Let $((l', r'), D) := ((l_1[p \leftarrow r_0\xi], r_1), C_0\xi\, C_1)\sigma$ and $\hat{t} := l_1\sigma$. Now we have: $((l', r'), D)\tau = ((l_1[p \leftarrow r_0\xi], r_1), C_0\xi\, C_1)\varrho = ((l_1\mu_1[p \leftarrow r_0\mu_0], r_1\mu_1), C_0\mu_0\, C_1\mu_1) = ((t_0, t_1), C_0\mu_0\, C_1\mu_1)$. Hence $D\tau$ is fulfilled. Furthermore, by $\hat{t}\tau = l_1\varrho = l_1\mu_1 = s$ and induction hypothesis we get $\forall u \lhd \hat{t}\tau$: $\Longrightarrow$ is confluent below u . If $t_0 = t_1$, then $t_0 \downarrow t_1$ trivially. Otherwise $(((l', r'), D), \hat{t}, p) \in \mathrm{CP}(\mathrm{R})$ and then by (*2*) we get $t_0 \downarrow t_1$, too.

Q. e. d. (The critical peak case; (*2*) ⇒ (*1*))

Proof of Theorem 8.18
Since the analoguos conclusions of Lemma 8.15 still hold, the proof reads just like the proof of Theorem 8.17 with the exception of the "Proof of Claim" in "The inductive case" of " (2) $\Rightarrow$ (1) ", where after the first sentence we have to add the fol.:

Since we have free constructors and $x\mu_1 \Longrightarrow x\nu$, we know $x \in \mathrm{V_{SIG}}$. Now, if $\mathcal{V}(L) \subseteq \mathrm{V_{CONS}}$, we get $x \notin \mathcal{V}(L)$, and thus $L\nu = L\mu_1$ is fulfilled. Therefore we can assume $\mathcal{V}(L) \not\subseteq \mathrm{V_{CONS}}$ in the sequel.

Proof of Lemma 8.19
Item (*1*) follows from item (*2*). By Lemma 8.15 we can now show items (*2*) and (*3*) by noetherian induction on s w. r. t. $\rhd$.
Proof of (*2*): There is only a finite number of of positions $p \in \mathcal{O}(s)$ and of rules $l{=}r \longleftarrow C$ of R matching s/p, and the set of matching substitutions $\sigma \in \mathcal{SUB}(\mathcal{V}(l{=}r \longleftarrow C), \mathcal{T}(\mathrm{X}))$ with $s/p = l\sigma$ is enumerable. Since $s \rhd s[p \leftarrow r\sigma]$ for $C\sigma$ being fulfilled, by induction hypothesis we only have to be able to semi-decide whether $C\sigma$ is fulfilled. In case of left-right-compatibility we semi-decide the fulfilledness of the literals from left to right; in case of compatibility only, we wait with our parallel enumeration until we have been able to establish the condition terms to be $\lhd$-smaller than $l\sigma$. Thus, for semi-deciding the fulfilledness of a literal L in $C\sigma$ we can assume its terms to be $\lhd$-smaller than s. By induction hypothesis it is obvious now how to semi-decide fulfilledness of '='- and 'Def'-literals in $C\sigma$, and how to semi-decide '$\neq$'-literals using item (*3*).
Proof of (*3*): By Lemma 6.8, the constructor rules being extra-variables free, and R and $\mathcal{O}(s)$ being finite, there can be only a finite number of t with $s \Longrightarrow t$, and by the induction hypothesis this also holds for those t with $s \overset{\oplus}{\Longrightarrow} t$. In case of left-right-compatibility we decide the fulfilledness of the literals from left to right; in case of compatibility only, we know that $C\sigma$ cannot be fulfilled if the test fails whether all its terms are $\lhd$-smaller than $l\sigma$. Thus, for deciding the fulfilledness of a literal L in $C\sigma$ we can assume its terms to be $\lhd$-smaller than s. By induction hypothesis it is obvious now how to decide fulfilledness of '='- and 'Def'-literals in $C\sigma$.

Proof of Lemma 8.20
Take the unconditional rule system of the proof of Lemma 8.1. From this system we are now going to construct our positive-conditional rule system by adding as new first argument a step-counter to our Turing machine by exchanging the recursive T-rules of the form " T l a r c s p = T l′ a′ r′ c′ s′ p′ " for rules of the form " T s x l a r c s p = T x l′ a′ r′ c′ s′ p′ " and the non-recursive T-rule by T x l a r stop s p = $\bot$. Finally we add the rule

$$\mathsf{terminatesp}\ l\ a\ r\ c\ s\ p = \mathsf{true} \longleftarrow \mathrm{T}\ x\ l\ a\ r\ c\ s\ p = \bot\ ,$$

where 'terminatesp' and 'true' are of a new sort. For the decidable ordering $\rhd$ of the termination-pair take the lexicographic path ordering where 'terminatesp' is bigger than all other function symbols and the variables of the old sorts; 'T' is bigger than all function symbols except 'terminatesp'; and 'cmd' and 'state' are bigger than 'nth'.
Now, since " terminatesp l a r c s p " is reducible iff the Turing machine of the proof of Lemma 8.1 halts, reducibility of ground terms cannot be co-semi-decidable.

Proof of Lemma 8.21
The proof is very similar to that of Lemma 8.19. Finally, (*3*) follows from (*2*), Theorem 8.17, and $\mathcal{T}(\mathrm{sig}, \mathrm{X})$ being enumerable.

Proof of Lemma 8.22
Take the unconditional part of the rule system of the proof of Lemma 8.20. For getting our positive conditional CRS R, add the rules

$$\mathsf{terminatesp}\ x = \mathsf{false}$$
$$\mathsf{terminatesp}\ x = \mathsf{true} \quad \longleftarrow \mathrm{T}\ x\ \mathrm{l\ a\ r\ c\ s\ p} = \bot\ ,$$

where '**terminatesp**', '**false**', and '**true**' are of a new sort; and l, a, r, c, s, p are ground terms. For the decidable ordering $\rhd$ of the termination-pair take the lexicographic path ordering where '**terminatesp**' is bigger than all other function symbols and the variables of the old sorts; 'T' is bigger than all function symbols except '**terminatesp**'; and '**cmd**' and '**state**' are bigger than '**nth**'.
Now, the fol. statements are logically equivalent:

- $\Longrightarrow_{\mathrm{R,V}}$ is confluent.
- The critical peak $(((\mathsf{true},\mathsf{false}), \mathrm{T}\ x\ \mathrm{l\ a\ r\ c\ s\ p} = \bot), \mathsf{terminatesp}\ x,\ \emptyset)$ is joinable w. r. t. R, V.
- There is no term $t \in \mathcal{T}(\mathrm{sig}, \mathrm{V_{SIG}} \uplus \mathrm{V_{CONS}})$ with $\mathrm{T}\ t\ \mathrm{l\ a\ r\ c\ s\ p} \overset{\circledast}{\Longrightarrow}_{\mathrm{R,V}} \bot$.
- The Turing machine of the proof of Lemma 8.1 does not halt.

Proof of Lemma 9.7
Proof of (1): Since $\mathrm{K} \subseteq \mathrm{M}$.
Proof of (2): If no rule in R has a negative condition, then there is some $\mathcal{A}$ being free in M over V w. r. t. some κ. Then $\mathcal{A}$ is in K. Let $\mathcal{B} \in \mathrm{M}$ and $\mu \in \mathcal{SUB}(\mathrm{V}, \mathcal{B})$. There must be some $h :: \mathcal{A} \to \mathcal{B}$ with $\mu = \kappa h$. Since " $\longrightarrow \Delta$ " is type-B-inductively valid by assumption, there must be some atom $(u{=}v)$ or some atom $(\mathrm{Def}\, u)$ in Δ which is true w. r. t. $\mathcal{A}_\kappa$. Then we have $\mathcal{A}_\kappa(u) = \mathcal{A}_\kappa(v)$ or $\mathcal{A}_\kappa(u) \in \mathcal{A}(\mathrm{CONS}, s)$ for some $s \in \mathrm{S}$. Then we have $\mathcal{B}_\mu(u) = \mathcal{B}_{\kappa h}(u) = h(\mathcal{A}_\kappa(u))$ and then $\mathcal{B}_\mu(u) = h(\mathcal{A}_\kappa(v)) = \mathcal{B}_\mu(v)$ or $\mathcal{B}_\mu(u) \in \mathcal{B}(\mathrm{CONS}, s)$.

Proof of Lemma 9.9
Proof of (1): Let $\mathcal{A}$ be a CONS:cons-term-generated constructor-minimal model of R. Let $\chi \in \mathcal{SUB}(\mathrm{V}, \mathcal{A})$. By the Axiom of Choice there is some $\tau \in \mathrm{INDSUB}(\mathrm{V}, \mathrm{cons})$ with $\chi = \tau \mathcal{A}_\chi$.
Claim: For $A \in \mathrm{At}(\mathrm{sig}, \mathrm{V})$: A is true w. r. t. $\mathcal{A}_\chi$ iff $A\tau$ is true w. r. t. $\mathcal{A}_\chi$.
Proof of Claim: By the Substitution-Lemma(4.1) we get for $u \in \mathcal{T}$:

$$\mathcal{A}_\chi(u) = \mathcal{A}_{\tau \mathcal{A}_\chi}(u) = \mathcal{A}_\chi(u\tau)$$ Q. e. d. (Claim)

Now, if each atom A in Γ is true w. r. t. $\mathcal{A}_\chi$, then (by Claim) $A\tau$ is true w. r. t. $\mathcal{A}_\chi$, too. Thus (since (Γ, Δ) is type-B-inductively valid by assumption), there must be some atom A in Δ for which $A\tau$ is true w. r. t. $\mathcal{A}_\chi$. Then (by Claim) A is true w. r. t. $\mathcal{A}_\chi$, too.
Proof of (2): Let $\mathcal{A}$ be a constructor-minimal model of R. Define the sig/cons-algebra $\mathcal{B}$ by $f^{\mathcal{B}} := f^{\mathcal{A}}$ $(f \in \mathrm{F})$; $\mathcal{B}(\mathrm{SIG}, s) := \mathcal{A}(\mathrm{SIG}, s)$ $(s \in \mathrm{S})$; $\mathcal{B}(\mathrm{CONS}, s) := \mathcal{A}[\mathcal{GT}_{\mathrm{CONS},s}]$ $(s \in \mathrm{S})$.
Claim 1: For $\chi \in \mathcal{SUB}(\mathrm{V}, \mathcal{B})$; $(u{=}v) \in \mathrm{At}(\mathrm{sig}, \mathrm{V})$:

$(u{=}v)$ is true w. r. t. $\mathcal{B}_\chi$ iff $(u{=}v)$ is true w. r. t. $\mathcal{A}_\chi$

Claim 2: For $\chi \in \mathcal{SUB}(\mathrm{V}, \mathcal{B})$; $(\mathrm{Def}\, u) \in \mathrm{At}(\mathrm{sig}, \mathrm{V})$:

If $(\mathrm{Def}\, u)$ is true w. r. t. $\mathcal{B}_\chi$, then $(\mathrm{Def}\, u)$ is true w. r. t. $\mathcal{A}_\chi$ too.

By these two trivial claims and $\mathcal{B} \lesssim_{\mathrm{H}} \mathcal{A}$; $\mathcal{B}$ is a constructor-minimal model of R, which, of course, is CONS:cons-term-generated. Let $\tau \in \mathrm{INDSUB}(\mathrm{V}, \mathrm{cons})$ and $\chi \in \mathcal{SUB}(\mathrm{V_{SIG}}, \mathcal{A}) = \mathcal{SUB}(\mathrm{V_{SIG}}, \mathcal{B})$. Now, suppose that for each atom A in Γ: $A\tau$ is true w. r. t. $\mathcal{A}_\chi$. Two cases: For $A = (u{=}v)$, by Claim 1 we know that $A\tau$ is true w. r. t. $\mathcal{B}_\chi$, and by the Substitution-Lemma(4.1) A is true w. r. t. $\mathcal{B}_{\tau\mathcal{B}_\chi}$. For $A = (\mathrm{Def}\, u)$, we know that " $\longrightarrow A$ " is type-C-inductively valid by assumption, i. e. A is true w. r. t. $\mathcal{B}_{\tau\mathcal{B}_\chi}$, too. Thus (since (Γ,Δ) is type-C-inductively valid by assumption), there must be some atom A in Δ which is true w. r. t. $\mathcal{B}_{\tau\mathcal{B}_\chi}$. By the Substitution-Lemma(4.1) $A\tau$ is true w. r. t. $\mathcal{B}_\chi$. By Claim 1 or by Claim 2 we know that $A\tau$ is true w. r. t. $\mathcal{A}_\chi$.

Proof of Lemma 9.12
Define $\mathcal{A} := \mathcal{T}(\mathrm{V_{SIG}})/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}}$. Define $\mathcal{I} := \mathcal{GT}/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\emptyset}$. Let κ be given by $(x \in \mathrm{V_{SIG}})$: $x \mapsto \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}}[\{x\}]$.
Proof of (1): By confluence of $\Longrightarrow_{\mathrm{R,V_{SIG}}}[\cap(D_{\mathrm{V_{SIG}}} \times D_{\mathrm{V_{SIG}}})]$ and Theorem 6.14(1), $\mathcal{A}$ is a constructor-minimum model of R. Furthermore, $\mathcal{A}$ is CONS:cons-term-generated.
Proof of (2): Let $\mathcal{B}$ be a CONS:cons-term-generated constructor-minimal model of R. Let $\mu \in \mathcal{SUB}(\mathrm{V},\mathcal{B})$. By confluence of $\Longrightarrow_{\mathrm{R},\emptyset}[\cap(D_\emptyset \times D_\emptyset)]$ and Theorem 6.14, there is a sig/cons-homomorphism $h :: \mathcal{A} \to \mathcal{B}$ with $\mu|_{\mathrm{V_{SIG}}} = \kappa h$. Furthermore, by the Axiom of Choice there is some $\tau \in \mathcal{SUB}(\mathrm{V_{CONS}},\mathcal{GT})$ with $\mu|_{\mathrm{V_{CONS}}} = \tau\mathcal{B} = \tau\mathcal{A}\, h$. Define $\chi \in \mathcal{SUB}(\mathrm{V},\mathcal{A})$ by $\chi := \kappa \uplus \tau\mathcal{A}$. We have $\mu = \chi h$. Now assume each atom A in Γ to be true w. r. t. $\mathcal{B}_\mu$. Two cases: If $A = (\mathrm{Def}\, u)$, then (since " $\longrightarrow A$ " is type-D-inductively valid by assumption) A is true w. r. t. $\mathcal{A}_\chi$. If $A = (u{=}v)$, then (since " $\longrightarrow \mathrm{Def}\, u$ " and " $\longrightarrow \mathrm{Def}\, v$ " are type-D-inductively valid by assumption) there are some $s \in \mathrm{S}$; $\hat{u}, \hat{v} \in \mathcal{GT}_{\mathrm{CONS},s}$ with $\mathcal{A}_\chi(u) = \mathcal{A}(\hat{u})$ and $\mathcal{A}_\chi(v) = \mathcal{A}(\hat{v})$. Then we get $\mathcal{B}(\hat{u}) = h_s(\mathcal{A}(\hat{u})) = h_s(\mathcal{A}_\chi(u)) = \mathcal{B}_{\chi h}(u) = \mathcal{B}_\mu(u) = \mathcal{B}_\mu(v) = \mathcal{B}_{\chi h}(v) = h_s(\mathcal{A}_\chi(v)) = h_s(\mathcal{A}(\hat{v})) = \mathcal{B}(\hat{v})$. By confluence of $\Longrightarrow_{\mathrm{R},\emptyset}[\cap(D_\emptyset \times D_\emptyset)]$ and Theorem 6.14(1), $\mathcal{I}$ and therefore also $\mathcal{B}$ is a constructor-minimum model. Hence (since $\hat{u}, \hat{v} \in \mathcal{GT}(\mathrm{cons})$) $\mathcal{I}(\hat{u}) = \mathcal{I}(\hat{v})$; i. e. $\hat{u} \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\emptyset} \hat{v}$. By Lemma 6.12 we get $\hat{u} \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} \hat{v}$, and then $\mathcal{A}_\chi(u) = \mathcal{A}(\hat{u}) = \mathcal{A}(\hat{v}) = \mathcal{A}_\chi(u)$. Thus A is true w. r. t. $\mathcal{A}_\chi$. All in all, all atoms in Γ are true w. r. t. to $\mathcal{A}_\chi$. By the assumed type-D-inductive validity of (Γ,Δ), there must be an atom A in Δ which is true w. r. t. $\mathcal{A}_\chi$. Two cases: If $A = (\mathrm{Def}\, u)$ with $u \in \mathcal{T}_{\mathrm{SIG},s}$, then (since $\mathcal{A}_\chi(u) \in \mathcal{A}(\mathrm{CONS},s)$) $\mathcal{B}_\mu(u) = \mathcal{B}_{\chi h}(u) = h_s(\mathcal{A}_\chi(u)) \in \mathcal{B}(\mathrm{CONS},s)$. If $B = (u{=}v)$, then (since $\mathcal{A}_\chi(u) = \mathcal{A}_\chi(v)$) $\mathcal{B}_\mu(u) = h_s(\mathcal{A}_\chi(u)) = h_s(\mathcal{A}_\chi(v)) = \mathcal{B}_\mu(v)$.

Proof of Lemma 9.13
Let $\mathcal{A} := \mathcal{T}(\mathrm{V_{SIG}})/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}}$. Let $\kappa \in \mathcal{SUB}(\mathrm{V_{SIG}},\mathcal{A})$ be given by $(x \in \mathrm{V_{SIG}})$: $x \mapsto \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}}[\{x\}]$.
Claim: For $\tau \in \mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{V_{SIG}}))$; $A \in \mathrm{At(sig,V)}$:

A is true w. r. t. $\mathcal{A}_{\tau\mathcal{A}_\kappa}$ iff $A\tau$ is true w. r. t. $\mathcal{A}_\kappa$.

Proof of Claim: By the Substitution-Lemma(4.1). Q. e. d. (Claim)

(1) ⇒ (2): Let $\tau \in \mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{V_{SIG}}))$ and suppose that for each $(u{=}v)$ in Γ we have $u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} v\tau$ and for each $(\mathrm{Def}\, u)$ in Γ we have some $\hat{u} \in \mathcal{GT}(\mathrm{cons})$ with $u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} \hat{u}$. Then for each atom A in Γ we know that $A\tau$ is true w. r. t. $\mathcal{A}_\kappa$, and then (by Claim) that A is true w. r. t. $\mathcal{A}_{\tau\mathcal{A}_\kappa}$. Thus (since (Γ,Δ) is type-D-inductively valid by assumption), there is some A in Δ which is true w. r. t. $\mathcal{A}_{\tau\mathcal{A}_\kappa}$. By Claim $A\tau$ is true w. r. t. $\mathcal{A}_\kappa$. Two cases: For $A = (u{=}v)$, we get $u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} v\tau$. For $A = (\mathrm{Def}\, u)$, we get some $s \in \mathrm{S}$ and some $\hat{u} \in \mathcal{GT}_{\mathrm{CONS},s} = \mathcal{T}(\mathrm{V_{SIG}})(\mathrm{CONS},s)$ with $u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} \hat{u}$.
(2) ⇒ (1): Let $\chi \in \mathcal{SUB}(\mathrm{V},\mathcal{A})$. Now suppose that each atom A in Γ is true w. r. t. $\mathcal{A}_\chi$. By the Axiom of Choice there is some $\tau \in \mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{V_{SIG}}))$ with $\chi = \tau\mathcal{A}_\kappa$. By Claim $A\tau$ is true w. r. t. $\mathcal{A}_\kappa$. Thus (since (2) holds by assumption), there is some A in Δ such that $A\tau$ is true w. r. t. $\mathcal{A}_\kappa$. By Claim, A is true w. r. t. $\mathcal{A}_\chi$.

Proof of Theorem 9.17

Let $\mathcal{A}'$ be a sig′/cons′-algebra. Define the sig/cons-algebra $\mathcal{A}$ by: $\mathcal{A} := \mathcal{A}'|_{\mathrm{F}\uplus\{\mathrm{SIG,CONS}\}\times\mathrm{S}}$. Now an "old" formula $(\Gamma,\Delta) \in \mathrm{Form(sig,V)}$ is valid in $\mathcal{A}$ iff it is valid in $\mathcal{A}'$. Furthermore, its inductive instances $(\Gamma,\Delta)\tau$ do not differ for $\tau \in \mathrm{INDSUB(V,cons)}$ and $\tau \in \mathrm{INDSUB(V',cons')}$ since $\forall s \in \mathrm{S} : \mathcal{GT}(\mathrm{cons})_s = \mathcal{GT}(\mathrm{cons'})_s$. Thus for proving (A) [and (B)], it suffices to show the fol.:

Claim: If $\mathcal{A}'$ is a [constructor-minimal] model of R′,
then $\mathcal{A}$ is a [constructor-minimal] model of R.

This Claim is also sufficient for (C); because if $\mathcal{A}'$ is CONS:cons′-term-generated, then $\mathcal{A}$ is CONS:cons-term-generated.

Proof of Claim: As each rule from R can be translated into an "old" formula (on which $\mathcal{A}$ and $\mathcal{A}'$ do not differ (cf. above)), $\mathcal{A}$ is a sig/cons-model of R. Let $\mathcal{GT}'$ denote the ground term algebra over sig′/cons′ and $\mathcal{GT}(\mathrm{cons})$ denote the ground term algebra over cons. By confluence of $\Longrightarrow_{\mathrm{R'},\emptyset}[\cap D'_\emptyset \times D'_\emptyset]$ and Theorem 6.14(1), $\mathcal{GT}'/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R'},\emptyset}$ 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. Therefore there must be some cons′-homomorphism $h' :: \mathcal{A}'|_{\mathrm{C'}\uplus(\{\mathrm{CONS}\}\times\mathrm{S'})} \to (\mathcal{GT}'/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R'},\emptyset})|_{\mathrm{C'}\uplus(\{\mathrm{CONS}\}\times\mathrm{S'})}$. By defining $h_s(a) := h'_s(a) \cap \mathcal{GT}(\mathrm{cons})$ ($a \in \mathcal{A}(\mathrm{CONS},s)$; $s \in \mathrm{S}$), we get a cons-homomorphism $h :: \mathcal{A}|_{\mathrm{C}\uplus(\{\mathrm{CONS}\}\times\mathrm{S})} \to \mathcal{GT}(\mathrm{cons})/(\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R'},\emptyset} \cap (\mathcal{GT}(\mathrm{cons})\times\mathcal{GT}(\mathrm{cons})))$. By confluence of $\Longrightarrow_{\mathrm{R'},\emptyset}[\cap D'_\emptyset \times D'_\emptyset]$ and Theorem 6.16(1) and Lemma 6.9 we get $(\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R'},\emptyset} \cap (\mathcal{GT}(\mathrm{cons})\times\mathcal{GT}(\mathrm{cons}))) \subseteq (\downarrow_{\mathrm{R'},\emptyset} \cap (\mathcal{GT}(\mathrm{cons})\times\mathcal{GT}(\mathrm{cons}))) \subseteq (\downarrow_{\mathrm{R},\emptyset} \cap (\mathcal{GT}(\mathrm{cons})\times\mathcal{GT}(\mathrm{cons}))) \subseteq \downarrow_{\mathrm{R},\emptyset,\omega} \subseteq \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R},\emptyset,\omega}$. Finally, for each sig/cons-model $\mathcal{C}$ of R we get $\forall\beta \preceq \omega : \forall s \in \mathrm{S} : \Longrightarrow_{\mathrm{R},\emptyset,\beta} \cap (\mathcal{GT}_{\mathrm{SIG},s}\times\mathcal{GT}_{\mathrm{SIG},s}) \subseteq \ker(\mathcal{C})_s$ by induction on β; thus $\forall s \in \mathrm{S} : (\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R'},\emptyset} \cap (\mathcal{GT}(\mathrm{cons})_s\times\mathcal{GT}(\mathrm{cons})_s)) \subseteq \ker(\mathcal{C})_s$; and then by the Homomorphism-Theorem(4.2) $\mathcal{A} \lesssim_{\mathrm{CONS}} \mathcal{C}$. Thus $\mathcal{A}$ is not only a sig/cons-model of R but also a constructor-minimal one. Q. e. d. (Claim)

Proof of (D): We do the proof for the equivalent version of type-D-inductive validity given by Lemma 9.13(2). Let $\tau' \in \mathcal{SUB}(\mathrm{V'},\mathcal{T}'(\mathrm{V'_{SIG}}))$ and suppose the premise of Lemma 9.13(2) to hold for this τ'. There is a $\tau \in \mathrm{INDSUB(V',cons')}$ and a $\sigma \in \mathcal{SUB}(\mathrm{V'_{SIG}},\mathcal{T}'(\mathrm{V'_{SIG}}))$ such that $\tau' = \tau\sigma$. Note that $\tau|_{\mathrm{V}} \in \mathcal{SUB}(\mathrm{V},\mathcal{T}(\mathrm{V_{SIG}}))$. For each atom $(u{=}v)$ in Γ we have assumed that " $\longrightarrow \mathrm{Def}\, u$ " and " $\longrightarrow \mathrm{Def}\, v$ " are valid in $\mathcal{T}(\mathrm{V_{SIG}})/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}}$. Thus there are $\hat{u},\hat{v} \in \mathcal{GT}(\mathrm{cons})$ with $u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} \hat{u}$ and $v\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} \hat{v}$. By Theorem 6.16(2) we get $u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R',V'_{SIG}}} \hat{u}$ and $v\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R',V'_{SIG}}} \hat{v}$, and then by Corollary 6.5 and the supposed $u\tau' \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R',V'_{SIG}}} v\tau'$ get $\hat{u} \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R',V'_{SIG}}} u\tau\sigma \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R',V'_{SIG}}} v\tau\sigma \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R',V'_{SIG}}} \hat{v}$. By Lemma 6.12 we get $\hat{u} \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R'},\emptyset} \hat{v}$ and then (by confluence of $\Longrightarrow_{\mathrm{R'},\emptyset}[\cap(D_\emptyset \times D_\emptyset)]$) $\hat{u} \downarrow_{\mathrm{R'},\emptyset} \hat{v}$, and then by Theorem 6.16(1) we get $\hat{u} \downarrow_{\mathrm{R},\emptyset} \hat{v}$ and then by Lemma 6.12 $u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} \hat{u} \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} \hat{v} \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} v\tau$. Furthermore, for each atom $(\mathrm{Def}\, u)$ in Γ we have assumed that " $\longrightarrow \mathrm{Def}\, u$ " is valid in $\mathcal{T}(\mathrm{V_{SIG}})/\overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}}$. Thus there is a $\hat{u} \in \mathcal{GT}(\mathrm{cons})$ with $u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} \hat{u}$. Thus (since (Γ,Δ) is type-D-inductively valid by assumption), there is a literal $(u{=}v)$ in Δ with $u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} v\tau$ or a literal $(\mathrm{Def}\, u)$ in Δ with $u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R,V_{SIG}}} \hat{u}$ for some $\hat{u} \in \mathcal{GT}(\mathrm{cons})$. By Theorem 6.16(2) we get $u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R',V'_{SIG}}} v\tau$ or $u\tau \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R',V'_{SIG}}} \hat{u}$, and then by Corollary 6.5 $u\tau\sigma \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R',V'_{SIG}}} v\tau\sigma$ or $u\tau\sigma \overset{\circledast}{\Longleftrightarrow}_{\mathrm{R',V'_{SIG}}} \hat{u}$.

B Unification

An (inefficient) algorithm for computing a most general unifier for a finite multi-set E of sort-invariant pairs of terms is given by the fol. inference system, which must be started with $(E,\ \mathcal{V}_{\mathrm{CONS}}(E,\mathrm{X}),\ \mathrm{id})$ for computing an element of $\mathrm{Mgu}(E,\mathrm{X})$ in the last position of any triple to which no inference rule applies anymore:

Delete:

$$\frac{(E \sqcup \langle (t,t)\rangle\ ,\mathrm{X}\ ,\sigma\ )}{(E\ ,\mathrm{X}\ ,\sigma\ )}$$

Function Symbol Clash:

$$\frac{(E \sqcup \langle (f s_0 \ldots s_{m-1}, g t_0 \ldots t_{n-1})\rangle\ ,\mathrm{X}\ ,\sigma\ )}{\mathrm{FAILURE}}$$

if $f \neq g$.

Divide:

$$\frac{(E \sqcup \langle (f s_0 \ldots s_m, f t_0 \ldots t_m)\rangle\ ,\mathrm{X}\ ,\sigma\ )}{(E \sqcup \langle\ (s_i,t_i) \mid i \leq m\ \rangle\ ,\mathrm{X}\ ,\sigma\ )}$$

Align:

$$\frac{(E \sqcup \langle (t,x)\rangle\ ,\mathrm{X}\ ,\sigma\ )}{(E \sqcup \langle (x,t)\rangle\ ,\mathrm{X}\ ,\sigma\ )}$$

if $(x \in \mathrm{V}\ \wedge\ t \notin \mathrm{V})$ or $(x \in \mathrm{V}_{\mathrm{SIG}}\ \wedge\ t \in \mathrm{V}_{\mathrm{CONS}})$.

Occur-In Clash:

$$\frac{(E \sqcup \langle (x,t)\rangle\ ,\mathrm{X}\ ,\sigma\ )}{\mathrm{FAILURE}}$$

if $x \in \mathcal{V}(t)$ and $x \neq t$.

Solve SIG-Variable:

$$\frac{(E \sqcup \langle (x,t)\rangle\ ,\mathrm{X}\ ,\sigma\ )}{(E\mu\ ,\mathrm{X}\ ,\sigma\mu\ )}$$

if $x \in \mathrm{V}_{\mathrm{SIG}} \setminus \mathcal{V}(t)$; $\mu \in \mathcal{SUB}(\mathrm{V},\mathcal{T})$ given by $x\mu = t$ and $\mu|_{\mathrm{V}\setminus\{x\}} = \mathrm{id}|_{\mathrm{V}\setminus\{x\}}$.

Solve CONS-Variable:

$$\frac{(E \sqcup \langle (x,t)\rangle\ ,\mathrm{X}\ ,\sigma\ )}{(E\mu\ ,\mathrm{X} \cup \mathrm{ran}(\xi)\ ,\sigma\mu\ )}$$

if $x \in \mathrm{V}_{\mathrm{CONS}} \setminus \mathcal{V}(t)$; $t \in \mathcal{T}(\mathrm{cons}, \mathrm{V}_{\mathrm{SIG}} \uplus \mathrm{V}_{\mathrm{CONS}})$; $\mu \in \mathcal{SUB}(\mathrm{V},\mathcal{T})$ given by $(y \in \mathrm{V})$

$$y\mu := \left\{ \begin{array}{ll} t(\xi \cup \mathrm{id}|_{\mathrm{V}_{\mathrm{CONS}}}) & \text{if } y = x \\ y\xi & \text{if } y \in \mathcal{V}_{\mathrm{SIG}}(t) \\ y & \text{otherwise} \end{array} \right\}$$

such that $\xi : \mathcal{V}_{\mathrm{SIG}}(t) \rightarrow \mathrm{V}_{\mathrm{CONS}} \setminus \mathrm{X}$ is injective.

Variable Sort Clash:

$$\frac{(E \sqcup \langle (x,t)\rangle\ ,\mathrm{X}\ ,\sigma\ )}{\mathrm{FAILURE}}$$

if $x \in \mathrm{V}_{\mathrm{CONS}}$ and $t \in \mathcal{T} \setminus \mathcal{T}(\mathrm{cons}, \mathrm{V}_{\mathrm{SIG}} \uplus \mathrm{V}_{\mathrm{CONS}})$.

Lemma B.1
The inference relation of the above inference system is noetherian on triples from

$$\mathrm{FMul}(\mathrm{DEq}(\mathrm{sig}, \mathrm{V}_{\mathrm{SIG}} \uplus \mathrm{V}_{\mathrm{CONS}})) \times \mathcal{F}(\mathrm{V}_{\mathrm{CONS}}) \times \mathcal{SUB}(\mathrm{V},\mathcal{T})\ .$$

If it terminates on $(E,\ \mathcal{V}_{\mathrm{CONS}}(E,\mathrm{X}),\ \sigma)$ *with* FAILURE, *then there is no* $\mu \in \mathcal{SUB}(\mathrm{V},\mathcal{T})$ *with* $\mathrm{set}[E]\mu \subseteq \mathrm{id}$. *If it terminates on* $(E,\ \mathcal{V}_{\mathrm{CONS}}(E,\mathrm{X}),\ \mathrm{id})$ *with* (E',X',σ), *then* $\sigma \in \mathrm{Mgu}(E,\mathrm{X})$, $\sigma\sigma = \sigma$, *and* $E' = \emptyset$.

Proof of Lemma B.1

Let '$\vdash$' denote the inference relation. $\vdash$ is noetherian because it either yields FAILURE or decreases (w. r. t. the lexicographic combination of (1.) the usual ordering on $\mathbb{N}$, (2.) the multi-set extension of the subterm ordering, (3.) the usual ordering on $\mathbb{N}$)
(E, X, σ) measured by
1. $|\mathcal{V}(E)|$,
2. $\langle\ \mathrm{set(e)}_0, \mathrm{set(e)}_1 \mid e \in E\ \rangle$,
3. $|\{\ e \in E \mid \mathrm{set(e)} \in ((\mathcal{T}\setminus \mathrm{V}) \times \mathrm{V}) \cup (\mathrm{V}_{\mathrm{CONS}} \times \mathrm{V}_{\mathrm{SIG}})\ \}|$.

Claim 1: Let $n \in \mathbb{N}$; $\forall i < n : (E_i, \mathrm{X}_i, \sigma_i) \vdash (E_{i+1}, \mathrm{X}_{i+1}, \sigma_{i+1})$; and $\sigma_0 = \mathrm{id}$. Now: $\forall \tau \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}) : ((\mathrm{set}[E_0]\sigma_n\tau \subseteq \mathrm{id}) \Leftrightarrow (\mathrm{set}[E_n]\tau \subseteq \mathrm{id}))$.
Proof of Claim 1: $n = 0$: Trivial. $n \Rightarrow (n+1)$: If $(E_n, \mathrm{X}_n, \sigma_n) \vdash (E_{n+1}, \mathrm{X}_{n+1}, \sigma_{n+1})$ by a Delete, Divide, or Align, the proof succeeds due to $\sigma_{n+1} = \sigma_n$ and $\mathrm{set}[E_0]\sigma_n\tau \subseteq \mathrm{id}$ iff $\mathrm{set}[E_n]\tau \subseteq \mathrm{id}$ iff $\mathrm{set}[E_{n+1}]\tau \subseteq \mathrm{id}$. Thus, suppose this step to be a Solve. Then for some E', using the denotation of the rules, we get $\sigma_{n+1} = \sigma_n\mu$; $E_n = E' \sqcup \langle (x,t) \rangle$; and $E_{n+1} = E'\mu$.
Sub-claim: $x\mu = t\mu$
Now: $\mathrm{set}[E_0]\sigma_{n+1}\tau \subseteq \mathrm{id}$ iff $\mathrm{set}[E_0]\sigma_n\mu\tau \subseteq \mathrm{id}$ iff (by induction hypothesis) $\mathrm{set}[E_n]\mu\tau \subseteq \mathrm{id}$ iff $(\mathrm{set}[E']\mu\tau \subseteq \mathrm{id} \wedge x\mu\tau = t\mu\tau)$ iff (by Sub-claim) $\mathrm{set}[E']\mu\tau \subseteq \mathrm{id}$ iff $\mathrm{set}[E_{n+1}]\tau \subseteq \mathrm{id}$.
Proof of Sub-claim: In case of a Solve SIG-Variable, we get $x\mu = t = t\mu$ using $x \notin \mathcal{V}(t)$ for the last step. In case of a Solve CONS-Variable, we get $x\mu = t(\xi \cup \mathrm{id}|_{\mathrm{V_{CONS}}}) = t\mu$ using $x \notin \mathcal{V}(t)$ for the last step. Q. e. d. (Claim 1)

Claim 2: Let $n \in \mathbb{N}$; $\forall i < n : (E_i, \mathrm{X}_i, \sigma_i) \vdash (E_{i+1}, \mathrm{X}_{i+1}, \sigma_{i+1})$; $\sigma_0 = \mathrm{id}$; and $\mathcal{V}_{\mathrm{CONS}}(E_0) \subseteq \mathrm{X}_0$. Now:
1. $\mathcal{V}_{\mathrm{CONS}}(E_n \cup \sigma_n[\mathrm{X}_0 \cup \mathrm{V}_{\mathrm{SIG}}]) \subseteq \mathrm{X}_n$
2. $\sigma_n|_{\mathcal{V}(E_n)\cup(\mathrm{V}_{\mathrm{CONS}}\setminus \mathrm{X}_n)} \subseteq \mathrm{id}$
3. $\sigma_n\sigma_n = \sigma_n$
4. If $\mathrm{set}[E_0]\pi_0 \subseteq \mathrm{id}$ for some $\pi_0 \in \mathcal{SUB}(\mathrm{V}, \mathcal{T})$, then there is some $\pi_n \in \mathcal{SUB}(\mathrm{V}, \mathcal{T})$ with $\pi_0|_{\mathrm{X}_0\cup \mathrm{V}_{\mathrm{SIG}}} = \sigma_n|_{\mathrm{X}_0\cup \mathrm{V}_{\mathrm{SIG}}}\pi_n$.

Proof of Claim 2(1): $n = 0$: Trivial. $n \Rightarrow (n+1)$: If $(E_n, \mathrm{X}_n, \sigma_n) \vdash (E_{n+1}, \mathrm{X}_{n+1}, \sigma_{n+1})$ by a Delete, Divide, or Align, the proof succeeds trivially. Now suppose this step to be a Solve CONS-Variable. Then $\mathcal{V}_{\mathrm{CONS}}(E_{n+1} \cup \sigma_{n+1}[\mathrm{X}_0 \cup \mathrm{V}_{\mathrm{SIG}}]) \subseteq \mathcal{V}_{\mathrm{CONS}}(\mu[\mathrm{V}_{\mathrm{SIG}} \cup \mathcal{V}_{\mathrm{CONS}}(E_n \cup \sigma_n[\mathrm{X}_0 \cup \mathrm{V}_{\mathrm{SIG}}])]) \subseteq \mathcal{V}_{\mathrm{CONS}}(\mu[\mathrm{V}_{\mathrm{SIG}} \cup \mathrm{X}_n]) \subseteq \mathrm{X}_n \cup \mathcal{V}_{\mathrm{CONS}}(t) \cup \mathrm{ran}(\xi) \subseteq \mathrm{X}_n \cup \mathrm{ran}(\xi) = \mathrm{X}_{n+1}$. For a Solve SIG-Variable we only have to omit the "$\cup\, \mathrm{ran}(\xi)$". Q. e. d. (Claim 2(1))

Proof of Claim 2(2): $n = 0$: Trivial. $n \Rightarrow (n+1)$: If $(E_n, \mathrm{X}_n, \sigma_n) \vdash (E_{n+1}, \mathrm{X}_{n+1}, \sigma_{n+1})$ by a Delete, Divide, or Align, the proof succeeds due to $\sigma_{n+1} = \sigma_n$, $\mathcal{V}(E_{n+1}) \subseteq \mathcal{V}(E_n)$, and $\mathrm{X}_{n+1} = \mathrm{X}_n$. Now suppose this step to be a Solve CONS-Variable. Since $\mathcal{V}(E_{n+1}) \subseteq \mathcal{V}(E_n\mu) \subseteq (\mathcal{V}(E_n) \setminus (\{x\} \cup \mathrm{dom}(\xi))) \cup (\mathrm{ran}(\xi) \setminus \{x\})$ (using $\mathrm{ran}(\xi) \cap \mathrm{X}_n = \emptyset$, $x \in \mathcal{V}_{\mathrm{CONS}}(E_n) \subseteq \mathrm{X}_n$ (by Claim 2(1)), $x \notin \mathcal{V}(t)$), it suffices to show $y\sigma_n\mu = y$ for the following three cases: First, for $y \in \mathcal{V}(E_n) \setminus (\{x\} \cup \mathrm{dom}(\xi))$ we get $y\sigma_n\mu = y\mu = y$. Second, for $y \in \mathrm{ran}(\xi) \setminus \{x\}$ by $y \in \mathrm{V}_{\mathrm{CONS}} \setminus \mathrm{X}_n$ we get $y\sigma_n\mu = y\mu = y$. Third, for $y \in \mathrm{V}_{\mathrm{CONS}} \setminus \mathrm{X}_{n+1} \subseteq \mathrm{V}_{\mathrm{CONS}} \setminus \mathrm{X}_n$ we have $y\sigma_n\mu = y\mu = y$ because if we had $y\mu \neq y$ then $y \in \mathcal{V}_{\mathrm{CONS}}(E_n) \subseteq \mathrm{X}_n$ (by Claim 2(1)). For a Solve SIG-Variable we only have to omit "$\mathrm{dom}(\xi)$", "$\mathrm{ran}(\xi)$", "$x \in \mathcal{V}_{\mathrm{CONS}}(E_n) \subseteq \mathrm{X}_n$" and the second case. Q. e. d. (Claim 2(2))

Proof of Claim 2(3): $n = 0$: Trivial. $n \Rightarrow (n+1)$: If $(E_n, \mathrm{X}_n, \sigma_n) \vdash (E_{n+1}, \mathrm{X}_{n+1}, \sigma_{n+1})$ by a Delete, Divide, or Align, the proof succeeds due to $\sigma_{n+1} = \sigma_n$. Now suppose this step to be a Solve.
Sub-claim: $\mu\sigma_n\mu = \sigma_n\mu$
Now we get $\sigma_{n+1}\sigma_{n+1} = \sigma_n\mu\sigma_n\mu = \sigma_n\sigma_n\mu = \sigma_n\mu = \sigma_{n+1}$.
Proof of Sub-claim:
Solve CONS-Variable: $x\mu\sigma_n\mu = t(\xi \cup \mathrm{id}|_{\mathcal{V}_{\mathrm{CONS}}(t)})\sigma_n\mu =$ (using Claim 2(2))
$t(\xi \cup \mathrm{id}|_{\mathcal{V}_{\mathrm{CONS}}(t)})\mu =$ (using $\mathrm{ran}(\xi) \cap \mathrm{X}_n = \emptyset$, $x \in \mathcal{V}_{\mathrm{CONS}}(E_n) \subseteq \mathrm{X}_n$ (by Claim 2(1)), $x \notin \mathcal{V}(t)$)
$t(\xi \cup \mathrm{id}|_{\mathcal{V}_{\mathrm{CONS}}(t)}) = x\mu =$ (using $x \in \mathcal{V}(E_n)$, Claim 2(2)) $x\sigma_n\mu$.
For $y \neq x$; $y \in \mathcal{V}_{\mathrm{SIG}}(t)$: $y\mu\sigma_n\mu = y\xi\sigma_n\mu =$ (using Claim 2(2))
$y\xi\mu =$ (using Claim 2(1) just as last time) $y\xi = y\mu =$ (using Claim 2(2)) $y\sigma_n\mu$.
For $y \neq x$; $y \in \mathrm{V} \setminus \mathcal{V}_{\mathrm{SIG}}(t)$: $y\mu\sigma_n\mu = y\sigma_n\mu$.
Solve SIG-Variable: Very similar. Q. e. d. (Claim 2(3))

Proof of Claim 2(4): $n = 0$: Trivial. $n \Rightarrow (n+1)$: If $(E_n, \mathrm{X}_n, \sigma_n) \vdash (E_{n+1}, \mathrm{X}_{n+1}, \sigma_{n+1})$ by a Delete, Divide, or Align, the proof succeeds due to $\sigma_{n+1} = \sigma_n$ by choosing $\pi_{n+1} := \pi_n$. Thus, suppose this step to be a Solve. Then for some E', using the denotation of the rules, we get $\sigma_{n+1} = \sigma_n\mu$; $E_n = E' \sqcup \langle (x,t) \rangle$; and $E_{n+1} = E'\mu$.
Sub-claim A: $t\pi_n = x\pi_n$
Sub-claim B: $\exists \pi_{n+1} \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}) : \mu|_{\mathrm{X}_n \cup \mathrm{V}_{\mathrm{SIG}}}\pi_{n+1} = \pi_n|_{\mathrm{X}_n \cup \mathrm{V}_{\mathrm{SIG}}}$
By induction hypothesis and for the next step by Claim 2(1) and Sub-claim B we succeed with $\pi_0|_{\mathrm{X}_0 \cup \mathrm{V}_{\mathrm{SIG}}} = \sigma_n|_{\mathrm{X}_0 \cup \mathrm{V}_{\mathrm{SIG}}}\pi_n = \sigma_n|_{\mathrm{X}_0 \cup \mathrm{V}_{\mathrm{SIG}}}\mu\pi_{n+1} = \sigma_{n+1}|_{\mathrm{X}_0 \cup \mathrm{V}_{\mathrm{SIG}}}\pi_{n+1}$.
Proof of Sub-claim B:
Solve CONS-variable: Define $\pi_{n+1} := (\xi^{-1} \cup \mathrm{id}|_{\mathrm{V}\setminus\mathrm{ran}(\xi)})\pi_n$. Now $\pi_{n+1} \in \mathcal{SUB}(\mathrm{V}, \mathcal{T})$ because sort-invariance is obvious and for $y \in \mathrm{V} \setminus \mathrm{ran}(\xi)$ we have $y\pi_{n+1} = y\pi_n$ and for $y \in \mathrm{ran}(\xi)$ we have $y\xi^{-1} \in \mathcal{V}(t)$, $y\xi^{-1}\pi_n \trianglelefteq_{\mathrm{ST}} t\pi_n = x\pi_n \in \mathcal{T}(\mathrm{cons}, \mathrm{V}_{\mathrm{CONS}})$ (using Sub-claim A, $x \in \mathrm{V}_{\mathrm{CONS}}$ and $\pi_n \in \mathcal{SUB}(\mathrm{V}, \mathcal{T})$), and then $y\pi_{n+1} = y\xi^{-1}\pi_n \in \mathcal{T}(\mathrm{cons}, \mathrm{V}_{\mathrm{CONS}})$.
Now: $x\mu\pi_{n+1} = t(\xi \cup \mathrm{id}|_{\mathcal{V}_{\mathrm{CONS}}(t)})(\xi^{-1} \cup \mathrm{id}|_{\mathrm{V}\setminus\mathrm{ran}(\xi)})\pi_n =$
(using $\mathrm{ran}(\xi) \cap \mathrm{X}_n = \emptyset$, $\mathcal{V}_{\mathrm{CONS}}(t) \subseteq \mathcal{V}_{\mathrm{CONS}}(E_n) \subseteq \mathrm{X}_n$ (by Claim 2(1)))
$t(\xi\xi^{-1} \cup \mathrm{id}|_{\mathcal{V}_{\mathrm{CONS}}(t)}\mathrm{id}|_{\mathrm{V}\setminus\mathrm{ran}(\xi)})\pi_n = t\pi_n = x\pi_n$ (using Sub-claim A).
For $y \neq x$, $y \in \mathcal{V}_{\mathrm{SIG}}(t)$, we have $y\mu\pi_{n+1} = y\xi\pi_{n+1} = y\pi_n$. For $y \neq x$, $y \in (\mathrm{X}_n \leftarrow \cup \mathrm{V}_{\mathrm{SIG}}) \setminus \mathcal{V}_{\mathrm{SIG}}(t)$, we have $y\mu\pi_{n+1} = y\pi_{n+1} = y\pi_n$ (using $\mathrm{ran}(\xi) \cap (\mathrm{X}_n \cup \mathrm{V}_{\mathrm{SIG}}) = \emptyset$).
Solve SIG-variable: Define $\pi_{n+1} := \pi_n$. Now: $x\mu\pi_{n+1} = t\pi_{n+1} = t\pi_n = x\pi_n$ (using Sub-claim A). For $y \neq x$ we have $y\mu\pi_{n+1} = y\pi_{n+1} = y\pi_n$. Q. e. d. (Sub-claim B)
Proof of Sub-claim A: By $\mathrm{set}[E_0]\sigma_n\pi_n = \mathrm{set}[E_0]\pi_0 \subseteq \mathrm{id}$ and Claim 1 we get $\mathrm{set}[E_n]\pi_n \subseteq \mathrm{id}$, i. e. $x\pi_n = t\pi_n$. Q. e. d. (Claim 2(4))

Claim 3: If $(E, \mathcal{V}_{\mathrm{CONS}}(E, \mathrm{X}), \sigma) \vdash^{\circledast} \mathrm{FAILURE}$, then $\neg\exists \mu \in \mathcal{SUB}(\mathrm{V}, \mathcal{T}) : \mathrm{set}[E]\mu \subseteq \mathrm{id}$.
Proof of Claim 3: W. l. o. g. suppose $(E, \mathcal{V}_{\mathrm{CONS}}(E, \mathrm{X}), \sigma) \vdash^{\circledast} (E', \mathrm{X}', \sigma'') \vdash \mathrm{FAILURE}$. Since the last element of the triple has no influence on the inference steps, there is a σ' with $(E, \mathcal{V}_{\mathrm{CONS}}(E, \mathrm{X}), \mathrm{id}) \vdash^{\circledast} (E', \mathrm{X}', \sigma') \vdash \mathrm{FAILURE}$. Now if we had $\mathrm{set}[E]\mu \subseteq \mathrm{id}$, then by Claim 2(4) there is some π with $\mathrm{set}[E]\sigma'\pi \subseteq \mathrm{id}$, and then by Claim 1 $\mathrm{set}[E']\pi \subseteq \mathrm{id}$, which contradicts $(E', \mathrm{X}', \sigma') \vdash \mathrm{FAILURE}$ due to the form of each of the inference rules yielding FAILURE. Q. e. d. (Claim 3)

Claim 4: If $(E, \mathrm{X}, \sigma) \notin \mathrm{dom}(\vdash)$, then $E = \emptyset$. (Trivial.)
Claim 5: If $(E, \mathcal{V}_{\mathrm{CONS}}(E, \mathrm{X}), \mathrm{id}) \vdash^{\circledast} (\emptyset, \mathrm{X}', \sigma)$, then $\sigma \in \mathrm{Mgu}(E, \mathrm{X})$ and $\sigma\sigma = \sigma$.
Proof of Claim 5: Since $\mathrm{set}[\emptyset]\,\mathrm{id} \subseteq \mathrm{id}$ we get $\mathrm{set}[E]\sigma \subseteq \mathrm{id}$ by Claim 1. For each $\mu \in \mathcal{SUB}(\mathrm{V}, \mathcal{T})$ with $\mathrm{set}[E]\mu \subseteq \mathrm{id}$ there is a $\pi \in \mathcal{SUB}(\mathrm{V}, \mathcal{T})$ by Claim 2(4) with $\mu|_{\mathrm{X}} = \sigma\pi|_{\mathrm{X}}$. Thus $\sigma \in \mathrm{Mgu}(E, \mathrm{X})$. Finally, by Claim 2(3) we get $\sigma\sigma = \sigma$. Q. e. d. (Claim 5)

References

[1] Jürgen Avenhaus, Klaus Becker. *Conditional Rewriting modulo a Built-in Algebra.* SEKI-Report SR-92-11, 1992. Fachbereich Informatik, Universität Kaiserslautern, D-67663 Kaiserslautern.

[2] Leo Bachmair, Nachum Dershowitz. *Commutation, Transformation, and Termination.* 8th CADE 1986, LNCS 230. Springer-Verlag, Berlin 1986.

[3] Leo Bachmair, Harald Ganzinger. *On Restrictions of Ordered Paramodulation with Simplification.* 10th CADE 1990, LNAI 449. Springer-Verlag, Berlin 1990.

[4] Leo Bachmair, Harald Ganzinger. *Perfect Model Semantics for Logic Programs with Equality.* Proc. of 8th Int. Conf. on Logic Programming, pp. 645-659. MIT Press, 1991.

[5] Leo Bachmair, Harald Ganzinger. *Rewrite-Based Equational Theorem Proving with Selection and Simplification.* Technical Report MPI-I-91-208. Max-Planck-Institut für Informatik, Saarbrücken, August 1991.

[6] Klaus Becker. *Semantics for Positive/Negative Conditional Rewrite Systems.* 3rd CTRS 1992, LNCS 656. Springer Verlag, Berlin 1993.

[7] Eddy Bevers, Johan Lewi. *Proof by Consistency in Conditional Equational Theories.* Report CW 102, Revised July 1990. Department of Computer Science, K. U. Leuven.

[8] Susanne Biundo, Birgit Hummel, Dieter Hutter, Christoph Walther. *The Karlsruhe Induction Theorem Proving System.* 8th CADE 1986, LNCS 230. Springer-Verlag, Berlin 1986.

[9] Adel Bouhoula, Emmanuel Kounalis, Michaël Rusinowitch. *Automated Mathematical Induction.* Technical Report 1636, INRIA, 1992.

[10] Robert S. Boyer, J Strother Moore. *A Computational Logic.* Academic Press, 1979.

[11] Nachum Dershowitz. *Termination of Rewriting.* J. Symbolic Computation (1987) 3, pp. 69-116.

[12] Nachum Dershowitz, Mitsuhiro Okada, G. Sivakumar. *Confluence of Conditional Rewrite Systems.* 1st CTRS 1988, LNCS 308. Springer-Verlag, Berlin 1988.

[13] Nachum Dershowitz, Mitsuhiro Okada, G. Sivakumar. *Canonical Conditional Rewrite Systems.* 9th CADE 1988, LNAI 310. Springer-Verlag, Berlin 1990.

[14] Harald Ganzinger, Jürgen Stuber. *Inductive Theorem Proving by Consistency for First-Order Clauses.* In: Informatik-Festschrift zum 60. Geburtstag von Günter Hotz. Teubner Verlag, Stuttgart 1992. Also in: 3rd CTRS 1992, LNCS 656. Springer Verlag, Berlin 1993.

[15] Martin Gogolla. *Algebraic Specifications with Partially Ordered Sorts and Declarations.* Forschungsbericht Nr. 169, 1983. Universität Dortmund, Postfach 500500, W-4600 Dortmund.

[16] Stéphane Kaplan. *Conditional Rewrite Rules.* Theoretical Computer Science 33 (1984), pp. 175-193. North-Holland.

[17] Stéphane Kaplan. *Positive/Negative Conditional Rewriting.* 1st CTRS 1988, LNCS 308. Springer-Verlag, Berlin 1988.

[18] Deepak Kapur, David R. Musser. *Proof by Consistency.* Artificial Intelligence 31 (1987), pp. 125-157.

[19] Deepak Kapur, David R. Musser. *Inductive Reasoning with Incomplete Specifications.* Preliminary Report. IEEE symposium on Logic In Computer Science 1986.

[20] Deepak Kapur, Paliath Narendran, Friedrich Otto. *On Ground-Confluence of Term Rewriting Systems.* Technical Report TR 87-6, Dept. of Computer Science, State Univ. of New York, USA, 1987.

[21] Emmanuel Kounalis, Michaël Rusinowitch. *On Word Problems in Horn Theories.* 9^{th} CADE 1988, LNAI 310. Springer-Verlag, Berlin 1988.

[22] Emmanuel Kounalis, Michaël Rusinowitch. *Mechanizing Inductive Reasoning.* Proc. of 8^{th} AAAI 1990. AAAI Press/The MIT Press, Menlo Park, Cambridge, London, Vol. 1, pp. 240-245.

[23] Peter Padawitz. *Horn Logic and Rewriting for Functional and Logic Program Design.* MIP-9002, März 1990. Fakultät für Mathematik und Informatik, Universität Passau, Postfach 2540, W-8390 Passau.

[24] David A. Plaisted. *Semantic Confluence Tests and Completion Methods.* Information and Control **65**, 1985, pp. 182-215.

[25] Gerd Smolka, Werner Nutt, Joseph A. Goguen, José Meseguer. *Order-Sorted Equational Computation.* In: H. Ait-Kaci, M. Nivat (eds.). *Resolution of Equations in Algebraic Structures, Vol. 2.* Academic Press, San Diego CA, 1989.

[26] Christoph Walther. *Argument-Bounded Algorithms as a Basis for Automated Termination Proofs.* 9^{th} CADE 1988, LNAI 310. Springer-Verlag, Berlin 1988.

[27] Christoph Walther. *Computing Induction Axioms.* LPAR 1992, LNAI 624. Springer-Verlag, Berlin 1992.

[28] Claus-Peter Wirth. *Inductive Theorem Proving in Theories specified by Positive/Negative Conditional Equations.* Diplomarbeit, 1991, Fachbereich Informatik, Universität Kaiserslautern, D-67663 Kaiserslautern.

[29] Claus-Peter Wirth, Bernhard Gramlich. *A Constructor-Based Approach for Positive/Negative-Conditional Equational Specifications.* SEKI-Report SR-92-10 (SFB), May 4, 1992, Fachbereich Informatik, Universität Kaiserslautern, D-67663 Kaiserslautern.

[30] Claus-Peter Wirth, Bernhard Gramlich. *A Constructor-Based Approach for Positive/Negative-Conditional Equational Specifications.* 3^{rd} CTRS 1992, LNCS 656. Springer Verlag, Berlin 1993.

[31] Hantao Zhang. *Reduction, Superposition and Induction: Automated Reasoning in an Equational Logic.* Rensselaer Polytech. Inst., Dept. of Comp. Sci., Troy, NY, PhD thesis, 1988.

[32] Hantao Zhang, Deepak Kapur, Mukkai S. Krishnamoorthy. *A Mechanizable Induction Principle for Equational Specifications.* 9^{th} CADE 1988, LNAI 310. Springer-Verlag, Berlin 1988.