Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 319 (2015) 315–331

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Sound and Complete Equational Reasoning over Comodels

Dirk Pattinson[1]

The Australian National University

Lutz Schröder [2,3]

Friedrich-Alexander-Universität Erlangen-Nürnberg

Abstract

Comodels of Lawvere theories, i.e. models in Set^{op}, model state spaces with algebraic access operations. Standard equational reasoning is known to be sound but incomplete for comodels. We give two sound and complete calculi for equational reasoning over comodels: an inductive calculus for equality-on-the-nose, and a coinductive/inductive calculus for equality modulo bisimulation which captures bisimulations syntactically through non-wellfounded proofs.

Keywords: Equational Logic, Comodels, Completeness, Bisimulation

1 Introduction

Comodels are an algebraic abstraction of the notion of global state, often used in the operational semantics of programming languages [13,15,11]. The most prominent example is the modelling of global state in imperative programs, where the explicit modelling of a store as a function that maps locations to values is replaced by algebraic operations that read and manipulate the values of global variables. Equations, in the standard universal-algebraic sense, ensure the intended semantics of these operations. Comodels are attractive for two reasons: first, they abstract implementation of state from the operational semantics, as state is not modelled explicitly, but only manipulated using operations. Second, the operations integrate seamlessly with programming language syntax. Some progress has been made towards the development of congruence formats in these settings [2]. While this builds

[1] dirk.pattinson@anu.edu.au

[2] lutz.schroeder@fau.de

[3] Work supported by the DFG under project COAX (SCHR 1118/12-1)

http://dx.doi.org/10.1016/j.entcs.2015.12.019

1571-0661/© 2015 Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

the link between operational and denotational semantics, the link with axiomatic semantics is much less understood. Comodels are essentially an algebraic concept, defined in terms of function symbols and equations, but the interpretation of function symbols takes place in the category Set^{op}, the opposite category of the category Set of sets and maps. While a unary function symbol, say wr_v for writing a value v, is still interpreted as a unary function $(\!|\mathsf{wr}_v|\!) : C \to C$ on a set (of comodel states), general n-ary function symbols are interpreted as functions $(\!|f|\!) : C \to n{\cdot}C = C + \dots + C$ (n times) and so are not understood as constructors, but as a combination of observation and state change. For example, a binary function symbol rd (that we think of as reading a binary value from a memory cell) receives the interpretation $(\!|\mathsf{rd}|\!) : C \to C + C$ and so indicates the value of the cell being read (by choosing one of the alternatives in $C + C$) on top of a new state. As a consequence, the standard tools of universal algebra for proving completeness of equational logic, including the construction of free algebras from terms modulo equations, are not available in the setting of comodels. Moreover, it is easy to see that equational reasoning is sound but incomplete over comodels. The easiest example is that of a theory comprising a nullary operation n and no equations: a comodel for this theory interprets n as a function $(\!|n|\!) : C \to 0{\cdot}C = \varnothing$ and therefore is empty, hence validates all equations; but clearly not all equations are derivable from the empty set of equations by standard equational reasoning. Excluding nullary operations does not improve this situation: we give an example below, due to Power, that shows that the same effect happens for a commutative binary operation.

This situation is remedied in the present paper, where we provide sound and complete calculi for equational reasoning over comodels. The overall flavour of comodels is coalgebraic, with a very simple type functor but with added complexity creeping in via the algebraic equations, which, for instance, may relate terms of different lookahead. The semantics of equations over comodels therefore naturally comes in two variants: satisfaction on-the-nose, and satisfaction up to bisimilarity, inducing correspondingly different notions of logical consequence. For reasoning on-the-nose, we give a standard, purely inductive calculus. We formalize this calculus in the style of a labelled sequent system. Key rules of the system express that terms with disjoint sets of free variables can never be equal in the comodel interpretation, and that terms with n free variables are essentially n-fold case statements allowing for a corresponding case distinction. For reasoning modulo bisimilarity, we then extend this inductive calculus by a single coinductive rule that allows us to conclude that two comodel terms are equal if they have the same output and their successors are equal. This rule may be applied in non-wellfounded proofs, resulting in a mixed inductive/coinductive calculus.

Related Work. We have already mentioned [14] where the theory of arrays is developed in terms of comodels, and the use of comodels in the semantics of programming languages [12,15,2]. None of these papers is concerned with axiomatic semantics, i.e. the equational logic of comodels. The model/comodel duality is investigated in [9,8] on the basis of clones and establishes, in our terminology, a dual equivalence between categories of comodels and certain topological spaces, but does not investigate the logical aspects. The proof-theoretic analysis of circular coinduction [16] has a goal that is similar to the completeness of equational reasoning modulo bisimulation, and blocks the application of the congruence rule in coinductive reasoning steps to achieve soundness. We achieve a similar effect by including substitution (which in our dualized setting plays, for purposes of coinductive proofs, an analogous role as congruence does in standard equational reasoning) as an axiom rather than a rule. As the proof calculus of *op.cit.* is purely inductive, no general completeness result can be established; this is remedied in our setup by using a mixed inductive-coinductive calculus. Mixed inductive / coinductive definitions have been investigated in type theory (e.g. [1,6]), and it appears that the modulo-bisimulation calculus given here can be straightforwardly encoded, thus presenting another example of the usefulness of these definition principles. An approach that is structurally similar to ours has been put forward for equational reasoning over non-wellfounded terms [7] where coinduction is used to capture non-wellfoundedness, whereas our calculus derives equations between finite terms, and employs coinduction to characterize validity modulo bisimulation. Non-wellfounded calculi have also been used in [5,4] to formalize arguments by infinite descent for inductive definitions (rather than up-to bisimulation arguments) where proofs are finite, possibly cyclic graphs subject to an external well-formedness condition.

2 Preliminaries and Notation

Categorical Notions. We write $+$ for (categorical) coproducts, and given $f_1, \dots, f_n : A_i \to B$ we write $[f_1, \dots, f_n] : A_1 + \dots + A_n \to B$ for the induced co-tuple. For a set V we write $V \cdot A = \coprod_{v \in V} A$ for the V-th copower of A, $\mathsf{inj}_v : A \to V \cdot A$ for the coproduct injection associated with $v \in V$, and (v, a) for the image $\mathsf{inj}_v(a)$ in the category of sets and functions. We identify every natural number $n \in \omega$ with the set of its predecessors, i.e. $n = \{0, \dots, n-1\}$. In particular, $\mathsf{inj}_i : A \to n \cdot A$ is the i-th coproduct injection for $i \in n$.

Algebraic Notions. A *signature* is a set Σ (of function symbols) equipped with a function $\mathsf{ar} : \Sigma \to \omega$ assigning arities to operations. We say that $f \in \Sigma$ is n-ary if $\mathsf{ar}(f) = n$. The set $T_\Sigma V$ of *Terms* over Σ with variables in V is defined in the standard way. If $t \in T_\Sigma(V)$ is a term, and f is unary, we often write $f.t$ for $f(t)$, and if $t(x)$ is a term with a free variable x, we write $t.x$ for $t(x)$. A *substitution* is a mapping $\sigma : V \to T_\Sigma(V)$; we write $t\sigma$ for the result of simultaneously replacing every free variable x in a term $t \in T_\Sigma(V)$ by $\sigma(x)$.

Fixpoints. If M is a monotone operator on a complete lattice (we only consider lattices of subsets in this paper), we write μM or $\mu X.M(X)$ for its least fixpoint, and νM, or $\nu X.M(X)$ for its greatest fixpoint.

3 Comodels

A comodel of an equational theory T is a model of the Lawvere theory L induced by T in the opposite category Set^{op} of the category Set of sets and functions. That is, a comodel is a finite coproduct preserving functor $\mathsf{L}^{\mathrm{op}} \to \mathsf{Set}$. In the area of programming language semantics, comodels are one way to explain the meaning of programs that change state. Given an algebraic signature Σ, a comodel for Σ consists of a carrier set, say C, that we think of as a set of states, and a function $(\!|f|\!) : C \to n \cdot C$ for every n-ary function symbol $f \in \Sigma$. As a consequence, a unary function symbol $f \in \Sigma$ is interpreted as a state-changing operation $(\!|f|\!) : C \to C$, whereas an n-ary function symbol g is an n-fold branching statement $(\!|g|\!) : C \to C + \cdots + C = n \cdot C$, for $n \geq 1$. That is, given a state $c \in C$, $(\!|g|\!)$ delivers a new state in one of n alternative branches. The presence of nullary function symbols (or constants) in a signature immediately implies that all comodels are empty, as constants are interpreted as functions $C \to 0 \cdot C = \emptyset$. In contrast to the evaluation of terms in universal algebra, the information flows from left to right when interpretation of terms over comodels. For example, when evaluating the term $g(f(x), h(y))$ over a given *algebra* A one first determines $a_0 = [\![f(x)]\!]$ and $a_1 = [\![h(y)]\!] \in A$ which then gives $[\![g(f(x), h(y))]\!] \in A$ by applying the interpretation of g to the pair (a_0, a_1). Interpreting the same term over a comodel C in state $c \in C$ *first* evaluates $(\!|g|\!)(c)$ to give $(i, c') \in 2 \cdot C$ and *then* applies (the interpretation of) $f(x)$ to c' in case $i = 0$, or the interpretation of $h(y)$ to c' if $c = 1$. The following definitions make this formal.

Definition 3.1 A *comodel* for an algebraic signature Σ, or Σ-*comodel* for short, is a tuple $(C, (\!|\cdot|\!))$ where C is a set, and $(\!|f|\!) : C \to n \cdot C$ is a function for all n-ary $f \in \Sigma$. Given a set V (of variables), every Σ-comodel engenders an interpretation of terms $(\!|\cdot|\!)^V : T_\Sigma V \to C \to V \cdot C$ by

$$(\!|v|\!)^V = \mathsf{inj}_v \text{ and } (\!|f(t_1, \ldots, t_n)|\!)^V = [(\!|t_1|\!)^V, \ldots, (\!|t_n|\!)^V] \circ (\!|f|\!)$$

where $T_\Sigma V$ denotes the set of Σ-terms with variables in V. If clear from the context, we will elide the superscript V. A *morphism* of comodels $(C, (\!|\cdot|\!)_C)$ and $(D, (\!|\cdot|\!)_D)$ is a function $h : C \to D$ that commutes with the interpretation of function symbols, i.e. $n \cdot h \circ (\!|f|\!)_C = (\!|f|\!)_D \circ h$ for all n-ary $f \in \Sigma$.

A comodel $(C, (\!|\cdot|\!))$ *satisfies* an equation $s = t$, where $s, t \in T_\Sigma V$ are terms over the set V of variables, if $(\!|s|\!)^V = (\!|t|\!)^V$, and if E is a set of equations, we say that $(C, (\!|\cdot|\!))$ is a Σ, E-comodel if it satisfies all equations in E.

A simple example of a comodel is the following one-bit memory cell that supports operations for reading and writing.

Example 3.2 Consider a one-bit memory cell, represented by comodels for the signature $\Sigma = \{\mathsf{rd}, \mathsf{wr}_0, \mathsf{wr}_1\}$ where wr_0 and wr_1 are unary and rd is binary. A comodel for Σ consists of a (state) set C and operations $(\!|\mathsf{wr}_0|\!), (\!|\mathsf{wr}_1|\!) : C \to C$ that we interpret as writing 0 (resp. 1) to the memory cell, and an operation $\mathsf{rd} : C \to C + C$ that will branch into the left hand component of the coproduct if evaluated at a cell storing 0, and into the right hand component, otherwise. Note that reading the cell may in general change its state. To ensure the intended behaviour of the memory cell, we stipulate the equations

$$E = \{\mathsf{wr}_0.\mathsf{rd}(x, y) = \mathsf{wr}_0.x, \mathsf{wr}_1.\mathsf{rd}(x, y) = \mathsf{wr}_1.y, \mathsf{rd}(x, x) = x\};$$

that is, the effect of reading a memory cell immediately after writing is completely determined by the bit written where the effect of writing to the cell is preserved, and reading the cell but ignoring the result is tantamount to doing nothing.

More examples of comodels and their theories, primarily concerned with state, can be found in [14,12].

As mentioned at the beginning of this section, comodels are usually presented as finite coproduct-preserving functors $\mathsf{L}^{\mathrm{op}} \to \mathsf{Set}$ where L is a Lawvere theory. For the purposes of this paper, it is more convenient to work with comodels for equational theories, as the latter determine concrete syntax that can be manipulated in the equational calculi that we are about to give. The equivalence of comodels for a Lawvere theory, and Σ, E-comodels is not relevant for the remainder of the paper, but included to justify our terminology. Recall that the Lawvere theory *induced by* a signature Σ and a set E of equations between Σ-terms is the category L whose objects are the natural numbers, and whose morphisms $\mathsf{L}^{\mathrm{op}}(n, m) = \mathcal{K}\mathsf{l}(UF)(n, m)$ are the morphisms in the Kleisli category of the monad UF where $U : \mathsf{Alg}(\Sigma, E) \to \mathsf{Set}$ is the forgetful functor and F its left adjoint.

Proposition 3.3 *Let E be a set of equations over a signature Σ. Then the category of Σ, E-comodels is isomorphic to the category of comodels for the Lawvere theory induced by Σ and E.*

Soundness of equational reasoning over comodels is an immediate corollary.

Corollary 3.4 *Let Σ be a signature and E a set of Σ-equations. If an equality $s = t$ is derivable from E in (standard) equational logic, then $(\!|s|\!) = (\!|t|\!)$ in every Σ, E-comodel $(C, (\!|\cdot|\!))$.*

4 Labelled Tableau Equality On-The-Nose

We proceed to give a complete system for deriving equalities between comodel terms. The following example due to Power shows that the usual proof systems of equational logic in general fail to be complete over comodels, and the main contribution of this paper is a complete calculus.

Example 4.1 (i) Let Σ be a signature that contains a nullary function symbol c. As pointed out at the beginning of Section 3, the only Σ-comodel is (carried by) the empty set. Therefore, irrespective of the set E of equations, all equations are valid over Σ-comodels, but not all equations are derivable in general, e.g. for $E = \varnothing$.

(ii) Consider the theory given by a single, commutative binary operation, i.e. $\Sigma = \{f\}$ with f binary, and $E = \{f(x, y) = f(y, x)\}$. If C is a comodel for Σ and E, then necessarily $C = \varnothing$: otherwise we could pick $c \in C$, and supposing that $(\!|f(x, y)|\!)(c) = \mathsf{inj}_x(d)$ we would obtain $\mathsf{inj}_x(d) = (\!|f(x, y)|\!)(c) = (\!|f(y, x)|\!)(c) = \mathsf{inj}_y(d)$ whence $x = y$ for distinct variables x and y, contradiction. As the supposition $(\!|f(x, y)(c) = \mathsf{inj}_y(d)$ leads to a similar contradiction, we obtain $C = \varnothing$ and as a consequence, C satisfies $s = t$ for all terms $s, t \in T_\Sigma(V)$. But clearly not all equations $s = t$ are derivable from E in equational logic (keeping algebraists off the dole).

The complete system we are about to introduce manipulates labelled expressions of the form $a.t = b.s$ where s, t are terms, and a, b are state variables, distinct from the variables from which we build terms. We interpret state variables a, b as elements of (the carrier of) the comodel, and read $a.s = b.t$ as saying that the terms s and t have the same denotation (on the nose) if evaluated in state a and b, respectively, of a given comodel. Throughout the section, we fix a signature Σ comprising terms with associated arities, a countable set V of (term) variables, and a countable set Z of *state variables*, disjoint from V. We write T for the set of terms built from function symbols in Σ using the variables in V, $\mathsf{FV}(t)$ for the set of (free) variables occurring in t, and $t(x_1, \ldots, x_n)$ to indicate that $\mathsf{FV}(t) \subseteq \{x_1, \ldots, x_n\}$.

Definition 4.2 A *labelled term* is a pair $(a, t) \in Z \times T$, written $a.t$, where t is a term and a is a state variable. A *labelled equation* is a pair of labelled terms, written $a.s = b.t$, with (free) state variables $\mathsf{FS}(a.s = b.t) = \{a, b\}$. A *(comodel) sequent* is a pair, written $\Gamma \Rightarrow A$, where Γ is a set of labelled equations, and A is a labelled equation. We briefly write A for $\emptyset \Rightarrow A$ and extend the notion of (free) state variables to sets of equations and comodel sequents by $\mathsf{FS}(\Gamma) = \bigcup\{\mathsf{FS}(A) \mid A \in \Gamma\}$ and $\mathsf{FS}(\Gamma \Rightarrow A) = \mathsf{FS}(\Gamma) \cup \mathsf{FS}(A)$. A *valuation* for a comodel $(C, (\!| \cdot |\!))$ is a function $\theta : Z \to C$, and we write $C, \theta \models a.s = b.t$ if $(\!|s|\!)(\theta a) = (\!|t|\!)(\theta b)$; $C, \theta \models \Gamma$ if $C, \theta \models B$ for all $B \in \Gamma$; and $C, \theta \models \Gamma \Rightarrow A$ if $C, \theta \models \Gamma$ implies $C, \theta \models A$. Finally, if E is a set of (unlabelled) equations, $E \models \Gamma \Rightarrow A$ if $C, \theta \models \Gamma \Rightarrow A$ for all comodel/valuation pairs C, θ where C satisfies all equations in E. A *renaming* is a function $\tau : Z \to Z$, and we write $(a.s = b.t)\tau$ for $\tau(a).s = \tau(b).t$ and $\Gamma\tau$ for $\{A\tau \mid A \in \Gamma\}$. Substitutions extend to labelled terms by $(a.t)\sigma = a.t\sigma$, to labelled equations by $(a.t = b.s)\sigma = (a.t\sigma = a.s\sigma)$, and to sets of labelled equations by $\Gamma\sigma = \{A\sigma \mid A \in \Gamma\}$.

We specifically do not require the antecedent Γ in a comodel sequent to be finite. In the calculus for equality on-the-nose, this enables us to speak about strong completeness, and we will later need infinite sets of assumptions in the calculus for equality modulo bisimulation.

Remark 4.3 State variables could be internalized in the original term language. A state variable is essentially a unary function symbol that only appears at the head of a term. In particular, this overloads the dot-notation for the application of unary functions $a.t = a(t)$ in a consistent way. All rules of the system we are about to introduce remain sound if we relax the interpretation of a state variable to be a unary function on states instead of a single state. (Completeness, proved later, transfers trivially to this more permissive semantics.) What distinguishes state variables from function symbols is that we have an infinite reservoir of state variables allowing us to pick fresh variables when necessary, as well as the possibility of renaming state variables.

As an equation $s = t$ is satisfied by all Σ, E-comodels if and only if $C, \theta \models a.s = a.t$ for all Σ, E-comodels C and valuations θ, it suffices to derive all valid comodel sequents.

The system that achieves this comprises the following rules. Substitution takes the form of an axiom

$$(\mathsf{subst})\frac{}{\Gamma, a.s(x_1,\dots,x_n) = b.t(x_1,\dots,x_n) \Rightarrow a.s(u_1,\dots,u_n) = b.t(u_1,\dots,u_n)}$$

where $u_1,\dots,u_n \in T_\Sigma V$ are terms, and stipulates that given that $a.s$ and $b.t$ take the same branches and end up in the same poststates, the same will hold if we postcompose with identical terms. Via the identity substitution, this implies in particular that every equation in Γ is derivable.

The fact that every term $r(x_1,\dots,x_n)$ evaluates to one of the alternatives x_1, $\dots$, x_n is captured by the rule

$$(\mathsf{case})\frac{\{\Gamma, a.r(x_1,\dots,x_n) = b.x_i \Rightarrow c.s = d.t \mid 1 \le i \le n\}}{\Gamma \Rightarrow c.s = d.t}(b \notin \mathsf{FS}(\Gamma \Rightarrow c.s = d.t)),$$

which employs similar mechanisms as disjunction elimination and existential elimination in natural deduction: to conclude $c.s = d.t$ we have to derive $c.s = d.t$ assuming each of the possible outputs x_i of some labelled term $a.r$ in turn, in each case giving a fresh name b to the poststate reached by $a.r$. We have a version of falsum-elimination (on the left) and a rule that asserts validity of all substitution instances of the axioms in E (on the right).

$$(\mathsf{disj})\frac{\Gamma \Rightarrow a.s = b.t}{\Gamma \Rightarrow c.u = d.v}(\mathsf{FV}(s) \cap \mathsf{FV}(t) = \varnothing) \qquad (E)\frac{}{\Gamma \Rightarrow a.s\sigma = a.t\sigma}(s = t \in E)$$

The falsum elimination rule (disj) (for *disjoint*) reflects the fact that labelled terms $a.s$ and $b.t$ that do not have any variable in common cannot be equal: if $(\!|s|\!)(\theta a) = (x, \alpha)$ and $(\!|t|\!)(\theta b) = (y, \beta)$ with $x \in \mathsf{FV}(s)$, $y \in \mathsf{FV}(t)$ then $x \neq y$ as $\mathsf{FV}(s) \cap \mathsf{FV}(t) = \varnothing$. From such an impossible equality, the rule therefore allows us to draw arbitrary conclusions $c.u = d.v$, in analogy to the classical ex-falso-quodlibet principle. Axiom (E) simply asserts that all substitution instances of equations are valid. These rules are completed with the standard rules for equality

$$(\mathsf{sym})\frac{\Gamma \Rightarrow a.s = b.t}{\Gamma \Rightarrow b.t = a.s} \quad (\mathsf{trans})\frac{\Gamma \Rightarrow a.s = b.t \quad \Gamma \Rightarrow b.t = c.u}{\Gamma \Rightarrow a.s = c.u} \quad (\mathsf{ref})\frac{}{\Gamma \Rightarrow a.t = a.t}$$

ensuring symmetry, reflexivity and transitivity of equality, and the renaming rule

$$(\mathsf{ren})\frac{\Gamma \Rightarrow A}{\Gamma\tau \Rightarrow A\tau}$$

as the only structural rule. We write $E \vdash \Gamma \Rightarrow A$ if $\Gamma \Rightarrow A$ can be derived using the above rules.

Remark 4.4 As the antecedent Γ of a comodel sequent $\Gamma \Rightarrow A$ may be infinite, an application of (case) could be blocked if Γ contains all state variables. The rule (ren) allows us to free a state variable that we can then use as a fresh variable in (case).

The next example revisits our examples of valid formulas not derivable in standard equational reasoning (Example 4.1), showing that the extended system does handle these examples.

Example 4.5 (i) If Σ contains a nullary function symbol, say n, we have $E \vdash a.n = a.n$ by (ref) and as $\mathsf{FV}(n) \cap \mathsf{FV}(n) = \emptyset$ we obtain $E \vdash a.s = b.t$ for arbitrary labelled terms $a.s$ and $b.t$ using (disj).

(ii) Consider a commutative binary function f, that is, $\Sigma = \{f\}$ and $E = \{f(x,y) = f(y,x)\}$. We show that $E \vdash c.s = d.t$ for all $c, d \in Z$ and all $s, t \in T$. Let $a \in Z$ be arbitrary and pick a fresh $b \in Z$, i.e. $b \notin \{a, c, d\}$. Fix $\Gamma_1 = \{a.f(x,y) = b.x\}$. We derive $E \vdash \Gamma_1 \Rightarrow c.s = d.t$, eliding the leading $E \vdash$, by

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_1 \Rightarrow a.f(x,y) = b.x}}{\Gamma_1 \Rightarrow b.x = a.f(x,y)} \qquad \dfrac{\dfrac{}{\Gamma_1 \Rightarrow a.f(x,y) = a.f(y,x)} \qquad \dfrac{}{\Gamma_1 \Rightarrow a.f(y,x) = b.y}}{\Gamma_1 \Rightarrow a.f(x,y) = b.y}}{\Gamma_1 \Rightarrow b.x = b.y}}{\Gamma_1 \Rightarrow c.s = d.t}$$

where the leftmost and rightmost leaves are by (subst) and we use (disj) in the last inference step. Taking $\Gamma_2 = \{a.f(x,y) = b.y\}$ we obtain $E \Rightarrow c.s = d.t$ by a symmetric derivation. We conclude, expanding the definitions of Γ_1 and Γ_2,

$$\dfrac{a.f(x,y) = b.x \Rightarrow c.s = d.t \qquad a.f(x,y) = b.y \Rightarrow c.s = d.t}{c.s = d.t}$$

using (case).

Soundness is proved by a standard induction over derivations.

Proposition 4.6 (Soundness) *The above system is sound for equality on-the-nose over comodels, i.e. $E \models \Gamma \Rightarrow A$ whenever $E \vdash \Gamma \Rightarrow A$.*

The following technical preparations are needed for establishing completeness. We formulate substitution as an axiom rather than as a rule, as the substitution axiom (but not the substitution rule, see Remark 5.13) remains sound also for modulo-bisimulation reasoning. However, substitution as a rule is admissible.

Lemma 4.7 (Admissibility of the Substitution Rule) *Suppose that $E \vdash \Gamma \Rightarrow a.s = b.t$ and let $\sigma : V \to T_\Sigma(V)$ be a substitution. Then $E \vdash \Gamma \Rightarrow a.s\sigma = a.t\sigma$.*

Moreover, cut is admissible.

Lemma 4.8 (Cut Admissibility) *Suppose that $E \vdash \Gamma \Rightarrow A$ and $E \vdash \Gamma, A \Rightarrow B$. Then $E \vdash \Gamma \Rightarrow B$.*

Although not strictly necessary for the technical development, we show that entailment is also closed under substitution for state variables.

Lemma 4.9 *Suppose that $E \vdash \Gamma \Rightarrow a.s_i = a.t_i$ for $i = 1, \dots, n$ and let $a \notin \mathsf{FS}(\Gamma)$. Then $E \vdash \Gamma \Rightarrow c.u(s_1, \dots, s_n) = c.u(t_1, \dots, t_n)$ for all n-ary function symbols $u \in \Sigma$.*

The completeness proof for the system is partly similar to the classical Henkin construction. We keep the set E of equations fixed throughout, and show that every non-derivable sequent has a countermodel. Since sequents can be infinite, we add additional state variables to obtain a Lindenbaum lemma.

Notation 4.10 We fix a second denumerable set Z' of state variables and call a state variable in $Z \cup Z'$ an *extended* state variable. We call a labelled term $a.s$ *extended* if $a \in Z \cup Z'$, and *standard* if $a \in Z$. An *extended* labelled equation is a labelled equation between extended terms, and a labelled equation between standard terms is called *standard.* We write $\vdash_{\mathsf{ext}}$ for derivability of extended labelled equations, and (continue to) write $\vdash$ for derivability of standard labelled equations.

We first establish that derivability in extended system is conservative. This is where we need renaming, as otherwise an application of (case)

$$\frac{\Gamma, a.s(x_1, \dots, x_n) = b.x_1 \Rightarrow A \quad \dots \quad \Gamma, a.s(x_1, \dots, x_n) = b.x_n \Rightarrow A}{\Gamma \Rightarrow A}$$

where $b \in Z'$ and Γ contains all state variables in Z couldn't be translated back to the standard system. Conservativity follows from the following:

Lemma 4.11 *Suppose that $E \vdash_{\mathsf{ext}} \Gamma \Rightarrow A$ and $\tau : Z \cup Z' \to Z$ is a bijective renaming. Then $E \vdash \Gamma\tau \Rightarrow A\tau$.*

Corollary 4.12 (Conservativity) *Suppose that $E \vdash_{\mathsf{ext}} \Gamma \Rightarrow A$ and Γ, A are standard. Then $E \vdash \Gamma \Rightarrow A$.*

The countermodel construction will be based on witnessed sets of labelled equations.

Definition 4.13 A set Γ of extended labelled equations is *witnessed* if for all extended labelled terms $a.s$ there exists a variable $x \in V$ and an extended state variable $b \in Z'$ such that $a.s = b.x \in \Gamma$.

Lemma 4.14 (Lindenbaum lemma) *Let Γ be a set of standard labelled equations, and A be a standard labelled equation such that $E \nvdash \Gamma \Rightarrow A$. Then Γ can be extended to a witnessed set $\hat{\Gamma}$ of (extended) labelled equations that is maximal with the property that $E \nvdash_{\mathsf{ext}} \hat{\Gamma} \Rightarrow A$.*

Maximal sets are closed under derivation:

Lemma 4.15 (Derivability is containment) *Let Γ be a set of (extended) labelled equations that is maximal with respect to $E \nvdash_{\mathsf{ext}} \Gamma \Rightarrow c.s = d.t$. Then $E \vdash_{\mathsf{ext}} \Gamma \Rightarrow A$ iff $A \in \Gamma$.*

We now construct a countermodel from a witnessed set of extended labelled sequents as in the Lindenbaum lemma; the elements of the countermodel are equivalence classes of extended state variables. This construction is related to Henkin's completeness proof of first-order logic.

Lemma 4.16 *Let Γ be a witnessed set of labelled equations that is maximal with the property that $E \nvdash_{\mathsf{ext}} \Gamma \Rightarrow c.s = d.t$. Then the following hold.*

(i) *For all labelled terms $a.u$ there exists a unique $x \in \mathsf{FV}(u)$ and a (not necessarily unique) $b \in Z'$ such that $E \vdash_{\mathsf{ext}} \Gamma \Rightarrow a.u = b.x$.*

(ii) *The relation* $\sim$ *on the set* $Z \cup Z'$ *of extended state variables defined by*

$$a \sim b \quad \textit{iff} \quad E \vdash_{\mathsf{ext}} \Gamma \Rightarrow a.x = b.x \textit{ for some } x \in V$$

is an equivalence.

(iii) *Putting* $(\!| f |\!)[a]_\sim = \mathsf{inj}_i([b]_\sim)$ *iff* $a.f(x_1, \dots, x_n) = b.x_i \in \Gamma$ *yields a well-defined comodel structure on* $Z'/\!\sim$.

(iv) *For all terms* u *and all* $a \in Z'$ *we have* $(\!| u |\!)[a] = (x, [b])$ *iff* $E \vdash_{\mathsf{ext}} \Gamma \Rightarrow a.u = b.x$.

(v) *For the valuation* $\theta(a) = [a]$ *we have* $Z'/\!\sim, \theta \models a.u = b.v$ *iff* $E \vdash_{\mathsf{ext}} \Gamma \Rightarrow a.u = b.v$.

(vi) *The comodel* $Z'/\!\sim$ *satisfies all equations in* E.

The first three items are straightforward, and the fourth is by induction on u. The rule (case) is not used in the proof of Lemma 4.16; its role in the completeness proof is to enable the construction of witnessed sets, i.e. the Lindenbaum lemma. Completeness is an immediate consequence of the countermodel construction:

Corollary 4.17 (Completeness for reasoning on-the-nose) *Comodel reasoning is complete for on-the-nose equality over comodels, i.e. if* $E \models \Gamma \Rightarrow a.s = b.t$ *then* $E \vdash \Gamma \Rightarrow a.s = b.t$.

5 Labelled Tableaux for Equality modulo Bisimulation

We now extend the reasoning system for equality on-the-nose introduced in the previous section to a system for equational reasoning modulo bisimilarity. Consider the following example.

Example 5.1 Recall the signature Σ and equations E describing a one-bit memory cell, introduced in Example 3.2. We intuitively expect that a second write overwrites the first, i.e. the equations $\mathsf{wr}_a.\mathsf{wr}_b.x = \mathsf{wr}_b.x$ hold for $a, b \in \{0, 1\}$. However, a comodel may internally record additional information beyond the content of the cell, for example the number of times the cell has been written to. Consider, for example, $C = \{0,1\} \times \mathbf{N}$ with $(\!| \mathsf{wr}_a |\!)(c, n) = (a, n+1)$ for $a = 0, 1$ and $(\!| \mathsf{rd} |\!)(c, n) = (c, (c, n))$. Clearly $(\!| \mathsf{wr}_0.\mathsf{wr}_0.x |\!)(0, n) = (0, n+2) \neq (0, n+1) = (\!| \mathsf{wr}_0.x |\!)(0, n)$ for any $n \in \mathbf{N}$ so that $E \not\models \mathsf{wr}_0.\mathsf{wr}_0.x = \mathsf{wr}_0.x$. On the other hand, we cannot tell $\mathsf{wr}_0.\mathsf{wr}_0.x$ and $\mathsf{wr}_0.x$ apart by (repeatedly) applying operations to both and observing that they give rise to different alternatives, as they behave identically under rd.

This phenomenon of comodel states being different but observationally indistinguishable is entirely standard, and captured by the established notion of bisimilarity. We explicitly instantiate this concept to the particular notation of comodels. For simplicity of notation, we work with bisimilarity on a single comodel; in the comodel setting, this is without loss of generality as bisimilarity across two models is the same as bisimilarity in their coproduct.

Definition 5.2 Let C be a Σ-comodel. A relation $B \subseteq C \times C$ is a *comodel bisimulation* if, for all $(c, c') \in B$ and all n-ary $f \in \Sigma$ we have that $(\!| f |\!)(c) B_n (\!| f |\!)(c')$ where

$B_n = \{(i,c),(i,c') \mid cBc' \text{ and } 0 \leq i < n\}$. We say that two comodel states c, c' are *comodel bisimilar*, and write $c \approx c'$, if they are related by some comodel bisimulation.

For any two elements (c_1, c_2) to be bisimilar, they have to branch into the same alternative under any $f \in \Sigma$, producing elements that are again bisimilar. In Example 3.2 we obtain that two states are bisimilar if and only if they store the same bit.

Example 5.3 Let Σ and E be as in Example 3.2 above, and let C be a Σ-comodel that satisfies E. For $c \in C$ and $i \in \{0,1\}$ we write $c \xrightarrow{\mathsf{rd}} i$ if there exists $d \in C$ such that $(\!|\mathsf{rd}|\!) = (i,d)$, that is, c branches to the i-th alternative under rd. Then the relation B defined by cBc' iff ($c \xrightarrow{\mathsf{rd}} i \iff c' \xrightarrow{\mathsf{rd}} i$ for $i = 0,1$) is a comodel bisimulation: if cBc' we need to establish that

$$(\!|\mathsf{wr}_i|\!)(c) B_1 (\!|\mathsf{wr}_i|\!)(c') \text{ and } (\!|\mathsf{rd}|\!)(c) B_2 (\!|\mathsf{rd}|\!)(c')$$

for $i = 0,1$. The left hand relationship follows from the equation $\mathsf{wr}_i.rd(x_0,x_1) = \mathsf{wr}_i.x_i$, and the right hand relationship from the equation $\mathsf{rd}(x,x) = x$.

It is clear that unions of comodel bisimulations are again comodel bisimulations, so that the largest bisimulation on a comodel always exists, and coincides with comodel bisimilarity. We can quotient comodels by bisimilarity:

Lemma and Definition 5.4 Let C be a Σ-comodel. Then bisimilarity $\approx$ is an equivalence relation on C, and putting

$$(\!|f|\!)([c]_\approx) = \mathsf{inj}_i([d]_\approx) \quad \text{iff} \quad (\!|f|\!)(c) = \mathsf{inj}_i(d)$$

yields a well-defined comodel structure on $C/\!\approx$, the *bisimulation quotient* $C/\!\approx$ of C.

Continuing Example 5.3 we obtain that the bisimulation quotient of any comodel satisfies the equation $\mathsf{wr}_a.\mathsf{wr}_b.x = \mathsf{wr}_b.x$.

Example 5.5 Let C be a Σ-comodel that satisfies E where Σ and E are as in Example 3.2. Then B as described in Example 5.3 is easily seen to coincide with comodel bisimilarity $\approx$ on C. Hence, $C/\!\approx\; \cong \{0,1\}$ with $(\!|\mathsf{wr}_0|\!)(\gamma) = 0$, $(\!|\mathsf{wr}_1|\!)(\gamma) = 1$ and $(\!|\mathsf{rd}|\!)(\gamma) = \mathsf{inj}_\gamma(\gamma)$. In particular, $C/\!\approx\; \models \mathsf{wr}_a.\mathsf{wr}_b.x = \mathsf{wr}_b.x$ for all $a, b \in \{0,1\}$.

We now extend the labelled deduction system given in the previous paragraph to account for equality modulo comodel bisimulation. We formalize this semantically by replacing on-the-nose satisfaction of (labelled) equations in a comodel by satisfaction of (the same) equations in its bisimulation quotient.

Definition 5.6 Let C be a Σ-comodel and $\theta : Z \to C$ a valuation. We write $C, \theta \models_\approx a.s = b.t$ if $C/\!\approx, \hat{\theta} \models a.s = b.t$ where $\hat{\theta}(a) = [a]_\approx$ (and $C/\!\approx$ is the bisimulation quotient of C). This extends to comodel sequents so that $C, \theta \models_\approx \Gamma \Rightarrow A$ iff $C, \theta \models_\approx A$ whenever $C, \theta \models_\approx B$ for all $B \in \Gamma$. We say that a comodel sequent is *valid up to bisimilarity* in the class of all comodels that satisfy E up to bisimilarity, in symbols $E \models_\approx \Gamma \Rightarrow A$, if $(C, \theta) \models_\approx \Gamma \Rightarrow A$ whenever $C/\!\approx$ is a Σ, E-comodel and θ is a valuation.

We now extend the derivation system $E \vdash$ to a derivation system $E \mathrel{|\!\sim}$ on comodel sequents to capture validity of equations modulo bisimulation. The ensuing system is a mixed inductive/coinductive proof system where the rules of $E \vdash$ may be applied *inductively*, and the following bisimulation rule

$$(\mathsf{bisim})\,\frac{\{\Gamma \Rightarrow a.f(x_1,\dots,x_n) = b.f(x_1,\dots,x_n) \mid n \in \mathbf{N}, f \in \Sigma\ n\text{-ary}\}}{\Gamma \Rightarrow a.x = b.x}$$

is applied *coinductively*. Formally, we partition the set of rules into

- *inductive* rules comprising (subst), (case), (disj), (sym), (ref), (trans), (ren) and (E) and say that $\Gamma \Rightarrow A$ is an *inductive consequence* of a set I of comodel sequents if it is the conclusion of an inductive rule with premises in I, and
- one *coinductive* rule (bisim), and say that $\Gamma \Rightarrow A$ is a *coinductive consequence* of a set C of comodel sequents if it is a conclusion of (bisim) with premises in C.

We write $\mathsf{Ind}(I)$ and $\mathsf{Coind}(C)$ for the set of inductive / coinductive consequences of sets I and C of comodel sequents, respectively. We then take *behavioural derivability* $E \mathrel{|\!\sim} \subseteq S$ to be the mixed fixpoint

$$E \mathrel{|\!\sim}\ = \nu C.\,\mu I.\,(\mathsf{Ind}(I) \cup \mathsf{Coind}(C))$$

where we write $E \mathrel{|\!\sim} \Gamma \Rightarrow A$ if $(\Gamma \Rightarrow A) \in E \mathrel{|\!\sim}$. That is, a sequent is behaviourally derivable if it is the conclusion of a possibly non-wellfounded derivation where the rule (bisim) can be applied infinitely often. The formulation of the proof system as a mixed fixpoint, with an inner inductive part, ensures that the inductive rules are only applied finitely many times between two successive instances of (bisim).

The soundness of the ensuing system is now no longer a simple matter of induction on derivations. Instead, we establish soundness in a step-by-step fashion using iterative approximations of behavioural derivability on the syntactic side, and iterative approximations of bisimilarity on the semantic side.

Notation 5.7 Let S be the set of comodel sequents, and consider the monotone operator $W : \mathcal{P}(S) \to \mathcal{P}(S)$, defined by

$$W(C) = \mu I.(\mathsf{Ind}(I) \cup \mathsf{Coind}(C))$$

and let $\mathrel{|\!\sim}_0 = \mathcal{P}(S)$ and $\mathrel{|\!\sim}_{n+1} = W(\mathrel{|\!\sim}_n)$. We say that $\Gamma \Rightarrow A$ is n-step derivable if $\mathrel{|\!\sim}_n \Gamma \Rightarrow A$. Similarly, for a comodel C, consider the monotone operator W_C : $\mathcal{P}(V \cdot C \times V \cdot C) \to \mathcal{P}(V \cdot C \times V \cdot C)$ defined by

$$W_C(R) = \{(x,c),(x,c') \mid (\!|f(x_1,\dots,x_n)|\!)(c) R (\!|f(x_1,\dots,x_n)|\!)(c') \text{for all } f \in \Sigma\}$$

for $R \subseteq V \cdot C \times V \cdot C$, and let $R_0 = (V \cdot C)^2$ and $R_{n+1} = W(R_n)$. We say that (v,c) and (v',c') are *n-step bisimilar* if $(v,c) R_n (v',c')$. If C is a comodel for Σ and $\Gamma \Rightarrow A$ is a comodel sequent, we say that $C,\theta \mathrel{|\!\approx}_n a.s = b.t$ if $(\!|a.s|\!) R_n (\!|b.t|\!)$, and $C,\theta \mathrel{|\!\approx}_n \Gamma \Rightarrow A$ if $C,\theta \mathrel{|\!\approx}_n A$ whenever $C,\theta \mathrel{|\!\approx} \Gamma$ (we explicitly require $C,\theta \mathrel{|\!\approx} \Gamma$, not just $C,\theta \mathrel{|\!\approx}_n \Gamma$). We say that $\Gamma \Rightarrow A$ is *n-step valid*, and write $E \mathrel{|\!\approx}_n \Gamma \Rightarrow A$, if $C,\theta \mathrel{|\!\approx}_n \Gamma \Rightarrow A$ for all Σ, E-comodels C and all valuations θ.

This allows us to show soundness by establishing that n-step derivability implies n-step validity. Bisimilarity is exhaustively approximated by n-step bisimilarity:

Lemma 5.8 *For states c, c' in a Σ-comodel C, $c \approx c'$ if and only if $(x, c) R_n (x, c')$ for all $n \in \mathbf{N}$ and all $x \in V$.*

It follows immediately that $\mathrel{|\!\approx}$ is exhaustively approximated by the $\mathrel{|\!\approx}_n$:

Lemma 5.9 *We have that $E \mathrel{|\!\approx} \; = \bigcap_{n \in \mathbf{N}} E \mathrel{|\!\approx}_n$.*

For behavioural derivability, one inclusion suffices.

Lemma 5.10 *We have that $E \mathrel{|\!\sim} \; \subseteq \bigcap_{n \in \mathbf{N}} E \mathrel{|\!\sim}_n$.*

The following is the key to soundness of behavioural derivability.

Lemma 5.11 *Let $n \in \mathbf{N}$ and let $\Gamma \Rightarrow A$ be a comodel sequent. Then $E \mathrel{|\!\sim}_n \Gamma \Rightarrow A$ implies that $E \mathrel{|\!\approx}_n \Gamma \Rightarrow A$.*

Proposition 5.12 *Let E be a set of Σ-equations. Then behavioural derivability is sound, i.e. $E \mathrel{|\!\approx} \Gamma \Rightarrow A$ whenever $E \mathrel{|\!\sim} \Gamma \Rightarrow A$.*

Proof. Let $E \mathrel{|\!\sim} \Gamma \Rightarrow A$. By Lemma 5.10, $E \mathrel{|\!\sim}_n \Gamma \Rightarrow A$ for all n, so by Lemma 5.11, $E \mathrel{|\!\approx}_n \Gamma \Rightarrow A$ for all n. By Lemma 5.9, it follows that $E \mathrel{|\!\approx} \Gamma \Rightarrow A$ □

Remark 5.13 The key to soundness, i.e. to Lemma 5.11, is that the substitution axioms (subst) are sound for $\mathrel{|\!\approx}_n$; the point is that $\mathrel{|\!\approx}_n$ requires the left-hand sides of comodel sequents to hold up to $\approx$ and not just up to $\approx_n$. In contrast, the substitution rule

$$(\mathsf{subst}')\dfrac{\Gamma \Rightarrow a.s(x_1, \dots, x_n) = b.t(x_1, \dots, x_n)}{\Gamma \Rightarrow a.s(u_1, \dots, u_n) = b.t(u_1, \dots, u_n)}$$

is sound for $\models$ and $\mathrel{|\!\approx}$ but not for $\mathrel{|\!\approx}_n$. And indeed, including this rule in the proof system for behavioural derivability is clearly unsound, as the following derivation (for $\Sigma = \{f_1, \dots, f_k\}$)would establish $a.x = b.x$ for arbitrary $a, b \in Z$ where $\overrightarrow{x}$ is a tuple of variables according to the arity of the preceding function symbol,

$$(\mathsf{bisim})\dfrac{(\mathsf{subst}')\dfrac{\dfrac{(\infty)}{a.x = b.x}}{a.f_1(\overrightarrow{x}) = b.f_1(\overrightarrow{x})} \qquad \cdots \qquad (\mathsf{subst}')\dfrac{\dfrac{(\infty)}{a.x = b.x}}{a.f_k(\overrightarrow{x}) = b.f_k(\overrightarrow{x})}}{a.x = b.x}$$

and (∞) indicates a coinductive repeat of the derivation. This would be a legal derivation in the inductive/coinductive format, as it uses only finitely many inductive rules between successive applications of (bisim); but of course $a.x = b.x$ should not be derivable for arbitrary a, b.

For completeness, we need to show that the sequents $\Gamma \Rightarrow A$ with $E \mathrel{|\!\approx} \Gamma \Rightarrow A$ are contained in the greatest fixpoint $\nu C.\mu I.(\mathsf{Ind}(I) \cup \mathsf{Coind}(C))$. If $V = \{\Gamma \Rightarrow A \mid E \mathrel{|\!\approx} \Gamma \Rightarrow A\}$ are the comodel sequents that are universally valid modulo bisimulation, this follows (using Knaster-Tarski) if $V \subseteq \mu I.(\mathsf{Ind}(I) \cup \mathsf{Coind}(V))$, that is, every $(\Gamma \Rightarrow A) \in V$ is inductively derivable from $\mathsf{Coind}(V)$. We follow the same approach as for completeness on-the-nose and use an additional, countable set of Henkin-constants.

Notation 5.14 As in 4.10, extend the set Z of state variables by a second, countable set Z' and consider labelled terms of the form $a.t$ where $a \in Z \cup Z'$ and $t \in T_\Sigma(V)$. As before, we call a labelled term (equation, set of equations) *standard* if they only mention state variables in Z, and *extended* otherwise.

We write $S = \{\Gamma \Rightarrow A \mid \Gamma \Rightarrow A \text{ standard}, E \models_{\approx} \Gamma \Rightarrow A\}$ and $X = \{\Gamma \Rightarrow A \mid \Gamma \Rightarrow A \text{ extended}, E \models_{\approx} \Gamma \Rightarrow A\}$ for the set of standard (resp. extended) comodel sequents that are universally valid modulo bisimulation. Finally, $E \vdash_S = \mu I.\mathsf{Ind}(I) \cup \mathsf{Coind}(S)$ is inductive entailment of standard labelled sequents from $\mathsf{Coind}(S)$, and $E {\vdash_{\mathsf{ext}}}_X = \mu I.\mathsf{Ind}(I) \cup \mathsf{Coind}(X)$ is inductive derivability of extended labelled sequents from $\mathsf{Coind}(X)$.

Using this notation, our proof strategy indicated above is formalized as follows.

Fact 5.15 *We have that ($E \mid\!\sim \Gamma \Rightarrow A$ whenever $\models_{\approx} \Gamma \Rightarrow A$) if $S \subseteq \mu I.(\mathsf{Ind}(I) \cup \mathsf{Coind}(S))$.*

That is, to show completeness for the mixed inductive/coinductive system, we have to show completeness for a modified inductive system where we may use $\mathsf{Coind}(S)$ as additional assumptions. We proceed as for equality-on-the nose, and re-visit the key lemmas.

Lemma 5.16 *Suppose $E {\vdash_{\mathsf{ext}}}_X \Gamma \Rightarrow A$ and $\tau : Z \cup Z' \to Z$ is a bijective renaming. Then $E \vdash_S \Gamma\tau \Rightarrow A\tau$.*

The proof of Corollary 4.12 translates directly to this new setting and we have:

Corollary 5.17 *Suppose that $E {\vdash_{\mathsf{ext}}}_X \Gamma \Rightarrow A$ and both Γ and A are standard. Then $E \vdash_S \Gamma \Rightarrow A$.*

Using the model construction of Lemma 4.16, we obtain the following.

Lemma 5.18 *Suppose that $E \nvdash_S \Gamma \Rightarrow c.s = d.t$. Then there exists a witnessed set $\hat{\Gamma}$ of (extended) labelled equations that is maximal with the property $E {\nvdash_{\mathsf{ext}}}_X \hat{\Gamma} \Rightarrow c.s = d.t$ and a comodel structure on $Z'/{\sim}$ where $a \sim b$ iff $a.x = b.x \in \Gamma$ for some $x \in V$ and a valuation θ defined by $\theta(a) = [a]_\sim$ such that*

(i) *$Z/{\sim}, \theta \models a.s = b.t$ iff $a.s = b.t \in \hat{\Gamma}$ and $Z'/{\sim}, \theta \not\models c.s = d.t$*

(ii) *If B is a comodel bisimulation on $Z/{\sim}$ then cBc' implies that $c = c'$*

The key step to showing that the model $Z/{\sim}$ only admits the diagonal as a bisimulation is to show that any equation A valid in $Z/{\sim}$ is in fact a behavioural consequence of $\hat{\Gamma}$, that is, $\hat{\Gamma} \Rightarrow A$ is universally valid modulo bisimulation. This allows us to make use of the additional assumptions, i.e. the coinductive consequences of universally valid sequents, to establish that A actually holds on-the-nose in $Z/{\sim}$. From the above, completeness is immediate:

Corollary 5.19 (Completeness modulo bisimulation) *Suppose that $E \models_{\approx} \Gamma \Rightarrow A$. Then $E \mid\!\sim \Gamma \Rightarrow A$.*

Recall from Example 5.5 that $E \not\models a.\mathsf{wr}_i.\mathsf{wr}_j.x = b.\mathsf{wr}_i.x$ under the axiomatization E defined in Example 3.2. We give an example derivation of this equation.

Example 5.20 Let Σ and E be as in Example 3.2. For $\alpha = a_0 \dots a_n \in \{0,1\}^+$ we write wr_α for $\mathsf{wr}_{a_0} \dots \mathsf{wr}_{a_n}$ and $\mathsf{wr}_{\alpha i}$ stands for $\mathsf{wr}_\alpha.\mathsf{wr}_i$. We show, generalizing the original goal, that $E \mathrel{|\!\!\sim} a.\mathsf{wr}_{\alpha i}.x = b.\mathsf{wr}_{\beta i}.x$ for all $\alpha, \beta \in \{0,1\}^*$, all $i = 0,1$, and all $a, b \in Z$. First note that for substitutions σ and τ, the rule

$$(\dagger) \quad \dfrac{\Gamma, a.s = c.u, b.t = d.v \Rightarrow a.s\sigma = b.t\tau}{\Gamma, a.s = c.u, b.t = d.v \Rightarrow c.u\sigma = d.v\tau}$$

is derivable using (subst), (sym) and (trans). We first show that $E \vdash \Gamma \Rightarrow a.\mathsf{wr}_{\alpha i}.\mathsf{rd}(x_0, x_1) = a.\mathsf{wr}_{\alpha i}.x_i$ for $i = 0,1$ and all sets Γ of comodel sequents (generally, one can, of course, show that weakening is admissible and then restrict to $\Gamma = \varnothing$), in the derivation $\mathcal{D}$

$$\dfrac{\dfrac{\dfrac{}{\Gamma, a.\mathsf{wr}_\alpha.x = c.x \Rightarrow c.\mathsf{wr}_i.\mathsf{rd}(x_0, x_1) = c.\mathsf{wr}_i.x_i}}{\Gamma, a.\mathsf{wr}_\alpha.x = c.x \Rightarrow a.\mathsf{wr}_{\alpha i}.\mathsf{rd}(x_0, x_1) = a.\mathsf{wr}_{\alpha i}.x_i}}{\Gamma \Rightarrow a.\mathsf{wr}_{\alpha i}.\mathsf{rd}(x_0, x_1) = a.\mathsf{wr}_{\alpha i}.x_i}$$

using (E), $(\dagger)$ and (case). We now demonstrate that $\Gamma \Rightarrow a.\mathsf{wr}_{\alpha i}.\mathsf{rd}(x_0, x_1) = b.\mathsf{wr}_{\beta i}.\mathsf{rd}(x_0, x_1)$ is inductively derivable from $\Gamma \Rightarrow a.\mathsf{wr}_{\alpha i}.x_i = b.\mathsf{wr}_{\beta i}.x_i$. Building on $\mathcal{D}$ and a variant $\mathcal{D}'$ obtained by replacing a with b, this is achieved, using symmetry and transitivity, by the derivation $\mathcal{E}$,

$$\dfrac{\dfrac{\dfrac{\dfrac{}{(\mathcal{D})}}{\Gamma \Rightarrow a.\mathsf{wr}_{\alpha i}.\mathsf{rd}(x_0, x_1) = a.\mathsf{wr}_{\alpha i}.x_i} \qquad \Gamma \Rightarrow a.\mathsf{wr}_{\alpha i}.x_i = b.\mathsf{wr}_{\beta i}.x_i}{\Gamma \Rightarrow a.\mathsf{wr}_{\alpha i}.x_i = b.\mathsf{wr}_{\beta i}.x_i} \qquad \dfrac{\dfrac{\dfrac{}{(\mathcal{D}')}}{\Gamma \Rightarrow b.\mathsf{wr}_{\beta i}.\mathsf{rd}(x_0, x_1) = b.\mathsf{wr}_{\beta i}.x_i}}{\Gamma \Rightarrow b.\mathsf{wr}_{\beta i}.x_i = b.\mathsf{wr}_{\beta i}.\mathsf{rd}(x_0, x_1)}}{\Gamma \Rightarrow a.\mathsf{wr}_{\alpha i}.\mathsf{rd}(x_0, x_1) = b.\mathsf{wr}_{\beta j}.\mathsf{rd}(x_0, x_1)}$$

We finally show that $\Gamma \Rightarrow a.\mathsf{wr}_{\alpha i}.x = b.\mathsf{wr}_{\beta i}.x$ is behaviourally derivable, writing

$$\Gamma_1 = \Gamma, a.\mathsf{wr}_{\alpha i}.x = c.x \quad \text{and} \quad \Gamma_2 = \Gamma_1, b.\mathsf{wr}_{\beta i}.x = d.x$$

to ease notation. If $c, d \notin \mathsf{FS}(\Gamma)$, we have the following derivation where (∞) indicates a coinductive repeat of the same derivation with evident modifications.

$$\dfrac{\dfrac{\dfrac{\dfrac{(\infty)}{\Gamma_2 \Rightarrow c.\mathsf{wr}_0.x_0 = d.\mathsf{wr}_0.x_0} \qquad \dfrac{(\infty)}{\Gamma_2 \Rightarrow c.\mathsf{wr}_1.x_0 = d.\mathsf{wr}_1.x_0} \qquad \dfrac{\dfrac{\dfrac{\dfrac{(\infty)}{\Gamma_2 \Rightarrow a.\mathsf{wr}_{\alpha i}.x_i = b.\mathsf{wr}_{\beta i}.x_i}}{(\mathcal{E})}}{\Gamma_2 \Rightarrow a.\mathsf{wr}_{\alpha i}.\mathsf{rd}(x_0, x_1) = b.\mathsf{wr}_{\beta i}.\mathsf{rd}(x, y)}}{\Gamma_2 \Rightarrow c.\mathsf{rd}(x_0, x_1) = d.\mathsf{rd}(x_0, x_1)}}{\Gamma_2 \Rightarrow c.x = d.x}}{\Gamma_2 \Rightarrow a.\mathsf{wr}_{\alpha i} = b.\mathsf{wr}_{\beta i}.x}}{\Gamma_1 \Rightarrow a.\mathsf{wr}_{\alpha i} = b.\mathsf{wr}_{\beta i}.x}}{\Gamma \Rightarrow a.\mathsf{wr}_{\alpha i} = b.\mathsf{wr}_{\beta i}.x}$$

The ternary inference is (bisim), followed by $(\dagger)$ and (case) (twice). This last proof is an infinite derivation where (bisim) is used infinitely often, but only a finite number of applications of the inductive rules are used between two successive applications of (bisim); that is, it fits our inductive/coinductive format. Thus, $E \mathrel{|\!\!\sim} \Gamma \Rightarrow a.\mathsf{wr}_{\alpha i}.x = b.\mathsf{wr}_{\beta i}.x$.

Generally, the way one will apply the inductive/coinductive calculus will be to identify a putative postfixed point, i.e. a set W of comodel sequents, and then show in an inductive proof that W can be derived from its coinductive consequences, i.e. that $W \subseteq \mu I. (\mathsf{Ind}(I) \cup \mathsf{Coind}(W))$. We have treated this principle informally in the above example; formally, we show in the example that the set W of labelled sequents $\Gamma \Rightarrow a.\mathsf{wr}_{\alpha i}.x = b.\mathsf{wr}_{\beta i}.x$, where Γ is any set of labelled equations, $a, b \in Z$, and $\alpha i, \beta i \in \{0,1\}^+$, is a postfixpoint.

Conclusions

We have given an inductive calculus for on-the-nose equational reasoning over comodels, and a mixed coinductive/inductive calculus for equational reasoning modulo bisimulation. We have done this in a bare bones setup without parametrized operations, e.g. using n *write* operations $\mathsf{wr}_0, \dots, \mathsf{wr}_{n-1}$ to modify memory cells that can store n distinct values, or one *print* operation for every character in a given character set. Similarly, reading (a character or memory location) is expressed by a function of arity equal to the number of alternatives. One natural extension would therefore be to include parametrized operations. An orthogonal direction of future research is to automate comodel reasoning in the style of the CIRC theorem prover [10]. This requires bisimulations to be either finite, or at least sufficiently well-structured to be analysed automatically. A second topic is to use complete reasoning over comodels to bridge between the operational/denotational and the axiomatic semantics of stateful programs.

References

[1] A. Abel. Mixed inductive/coinductive types and strong normalization. Asian Symposium on Programming Languages and Systems, APLAS 2007, LNCS, vol. 4807, pp. 286–301. Springer, 2007.

[2] F. Abou-Saleh and D. Pattinson. Comodels and effects in mathematical operational semantics. *Foundations of Software Science and Computations Structures, FoSSaCS 2013*, LNCS, vol. 7794, pp. 129–144. Springer, 2013.

[3] F. Borceux. *Handbook of Categorical Algebra.* Cambridge University Press, 1994.

[4] J. Brotherston, N. Gorogiannis, and R. L. Petersen. A generic cyclic theorem prover. Asian Symposium on Programming Languages and Systems, APLAS 2012, LNCS, vol. 7705, pp. 350–367. Springer, 2012.

[5] J. Brotherston and A. Simpson. Complete sequent calculi for induction and infinite descent. Logic in Computer Science, LICS 2007, pp. 51–62. IEEE Computer Society, 2007.

[6] N. Danielsson and T. Altenkirch. Mixing induction and coinduction. http://www.cse.chalmers.se/~nad/publications/danielsson-altenkirch-mixing.html, 2009. Unpublished manuscript.

[7] J. Endrullis, H. H. Hansen, D. Hendriks, A. Polonsky, and A. Silva. A coinductive treatment of infinitary rewriting. *CoRR*, abs/1306.6224, 2013.

[8] S. Kerkhoff. A general duality theory for clones. *Int. J. Alg. Comput.*, 23(3):457–502, 2013.

[9] S. Kerkhoff. Dualizing clones as models of lawvere theories. Algebraic Complexity Theory, WACT 2013, ENTCS, vol. 303, pp. 79105. Elsevier, 2014

[10] D. Lucanu, E. Goriac, G. Caltais, and G. Rosu. CIRC: A behavioral verification tool based on circular coinduction. Algebra and Coalgebra in Computer Science, CALCO 2009, LNCS, vol. 5728, pp. 433–442. Springer, 2009.

[11] R. Møgelberg and S. Staton. Linear usage of state. *Log. Meth. Comput. Sci.* 10, 2014.

[12] G. Plotkin and J. Power. Tensors of comodels and models for operational semantics. In *Mathematical Foundations of Programming Semantics, MFPS 2008*, ENTCS, vol. 218, pp. 295–311. Elsevier, 2008.

[13] G. Plotkin. Adequacy for algebraic effects with state. Algebra and Coalgebra in Computer Science, CALCO 2005, LNCS, vol. 3629, p. 51. Springer, 2005.

[14] J. Power and O. Shkaravska. From comodels to coalgebras: State and arrays. Coalgebraic Methods in Computer Science, CMCS 2004, ENTCS, vol. 106, pp. 297–314. Elsevier, 2004.

[15] J. Power. Semantics for local computational effects. Mathematical Foundations of Programming Semantics, MFPS 2006, ENTCS, vol. 158, pp. 355–371. Elsevier, 2006.

[16] G. Rosu and D. Lucanu. Circular coinduction: A proof theoretical foundation. Algebra and Coalgebra in Computer Science, CALCO 2009, LNCS, vol. 5728, pp. 127–144. Springer, 2009.