To appear in: Electronic Notes in Theoretical Computer Science 72 No. 1 (2005)
URL: http://www.elsevier.nl/locate/entcs/volume72.html 16 pages

A Duality in Proof Systems for Recursive Type Equality and for Bisimulation Equivalence on Cyclic Term Graphs

Clemens Grabmayer [1,2]

Department of Mathematics and Computer Science
Vrije Universiteit Amsterdam
de Boelelaan 1081a, 1081 HV Amsterdam,
The Netherlands

Abstract

This paper is concerned with a proof-theoretic observation about two kinds of proof systems for regular cyclic objects. It is presented for the case of two formal systems that are complete with respect to the notion of "recursive type equality" on a restricted class of recursive types in μ-term notation. Here we show the existence of an immediate duality with a geometrical visualization between proofs in a variant of the coinductive axiom system due to Brandt and Henglein and "consistency-unfoldings" in a variant of a 'syntactic-matching' proof system for testing equations between recursive types due to Ariola and Klop.

Finally we sketch an analogous result of a duality between a similar pair of proof systems for bisimulation equivalence on equational specifications of cyclic term graphs.

1 Introduction

The main part of this paper is concerned with an observation about two complete proof systems for the notion of "recursive type equality" on recursive types.

There are to our knowledge basically two different complete axiom systems known for recursive type equality: (i) A system due to R. Amadio and L. Cardelli given in [1] (1993) and (ii) a coinductively motivated axiom system introduced by M. Brandt and F. Henglein in [3] (1998). Apart from these axiomatizations, it is also possible to consider (iii) a 'syntactic-matching' proof system for which a notion of consistency with respect to this system is complete for recursive type equality. Such a system can be defined in a very similar way to one that has been introduced by

[1] Email: clemens@cs.vu.nl ; homepage: http://www.cs.vu.nl/~clemens .

[2] I want to thank J.W. Klop for suggesting this study and for providing me with a couple of initial ideas. And I am also much indebted to Bas Luttik for his careful reading of drafts for this paper.

Z. Ariola and J.W. Klop in [2] (1995) for the notion of bisimulation equivalence on equational representations of cyclic term graphs. For our purpose we will consider only 'normalized' variants without symmetry and transitivity rules of the Brandt–Henglein and syntactic-matching systems. In Section 3 these variant systems will be defined and their respective soundness and completeness theorems stated.

It was noted by J.W. Klop that there appears to be a striking similarity between the activities of (a) trying to demonstrate the consistency of an equation between recursive types with respect to the syntactic-matching system and of (b) trying to prove the same equation in the system of Brandt and Henglein. This basic observation underlying the present paper will be described in Section 4 in relation to the introduced variant systems by explaining it in the light of an example.

In order to extract a precise statement from this observation, two formal prerequisites turn out to be necessary: Firstly, in Section 5 we will introduce an extension of the variant Brandt–Henglein system with some more coinductive rules. And secondly, in Section 6 we define so called "consistency-unfoldings" of given equations between recursive types in the variant 'syntactic-matching' system as certain formalizations of successful consistency-checks. With these notions our main theorem is stated in Section 7: There exists even a "duality" between derivations in the variant Brandt–Henglein system and the corresponding consistency-unfoldings in the variant syntactic-matching system via easily definable reflection mappings.

In Section 8 we furthermore outline an analogous result for a similar pair of proof systems concerned with the bisimulation relation on equational specifications of cyclic term graphs.

2 Preliminaries on recursive types

As in Brandt and Henglein [3] we consider only recursive types denoted by μ-terms in canonical form over the restricted class of finite types with $\to$ as the single type constructor. We assume a countably infinite set $TVar$ of *type variables*. The small Greek letters α and β (possibly with subscripts) will be used as syntactical variables for type variables and the letters τ, σ, ρ, χ for recursive types.

Definition 2.1 (Recursive Types *can-μTp* in Canonical Form). The set *can-μTp* of *recursive types in canonical form* is generated by the following grammar:

$$\tau \;::=\; \bot \mid \top \mid \alpha \mid \tau_1 \to \tau_2 \mid \mu\alpha.\underbrace{(\tau_1 \to \tau_2)}_{\text{where } \alpha \in \mathrm{fv}(\tau_1 \to \tau_2)} \quad . \tag{2.1}$$

The set of all equations $\tau = \sigma$ between recursive types τ and σ in canonical form will be denoted by *can-μTp–Eq*.

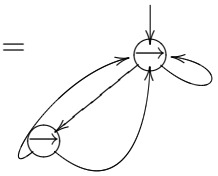
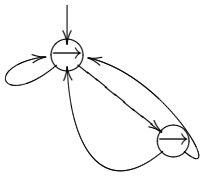
The recursive types in *can-μTp* are in "canonical form" due to the two requirements in the last disjunctive clause in grammar (2.1): For given $\alpha \in TVar$ the μ-operator may only be applied to a previously formed expression τ if τ is of the form $\tau_1 \to \tau_2$ and if α occurs free in $\tau_1 \to \tau_2$. – Our results do not depend on the limitation to consider recursive types *in canonical form* only (cf. forthcoming [4]).

Figure 1 Example of two strongly equivalent recursive types.

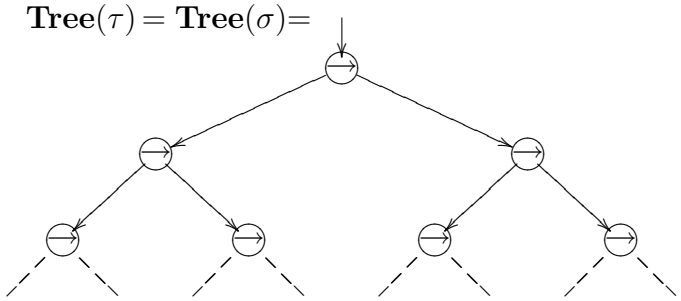
We consider the recursive types in canonical form

$$\tau :\equiv \mu\alpha.\,((\alpha \to \alpha) \to \alpha) \qquad \text{and} \qquad \sigma :\equiv \mu\alpha.\,(\alpha \to (\alpha \to \alpha)) \;.$$

These correspond respectively to the different cyclic term graphs

$G(\tau) =$  and $G(\sigma) =$  ,

but they possess the same tree unfolding of the form

$\mathbf{Tree}(\tau) = \mathbf{Tree}(\sigma) =$  .

Hence τ and σ are strongly equivalent, i.e. $\tau =_\mu \sigma$ holds, due to Definition 2.2.

Contrary to [3], we do not implicitly identify recursive types in *can-μTp* that can be obtained from each other by a finite sequence of admissible renaming-steps for bound type variables, i.e. that are *variants* of each other. We will use the notation $\tau_1 \equiv_{\mathrm{v}} \tau_2$ to express that τ_1 and τ_2 are variants of each other.

Via a natural transformation of μ-terms into cyclic term graphs described in (the extended version of) [2], it is possible to assign to every recursive type $\tau \in$ *can-μTp* a cyclic term graph $G(\tau)$ whose nodes have at most two outgoing edges and are labelled by either the binary function symbol $\to$ or by a symbol of arity zero in $\{\bot, \top\} \cup TVar$. Relying on this transformation, the *tree unfolding* $\mathbf{Tree}(\tau)$ of an arbitrary recursive type $\tau \in$ *can-μTp* can be defined as the tree unfolding of $G(\tau)$. An alternative formal definition of $\mathbf{Tree}(\tau)$ can be found in [1]. [3] The *leading symbol* $\mathcal{L}(\tau)$ of a recursive type $\tau \in$ *can-μTp* is defined as the symbol that labels the root in the tree unfolding $\mathbf{Tree}(\tau)$ of τ. [4]

Definition 2.2 (Recursive Type Equality (Strong Equivalence) $=_\mu$). Two recursive types $\tau, \sigma \in$ *can-μTp* are called *strongly equivalent* (symbolically denoted by: $\tau =_\mu \sigma$) iff they possess the same tree unfolding. More formally, the equivalence relation *recursive type equality* (also called *strong recursive type equivalence*) $=_\mu$ is

[3] The definition in [1] is slightly more general than then the one needed here because Amadio and Cardelli allow also recursive types *not in canonical form* like for example $\mu\alpha.\,(\mu\beta.\,\alpha)$.

[4] Alternatively and more formally $\mathcal{L}(\tau)$ can be defined for all $\tau \in$ *can-μTp* by the 5 clauses $\mathcal{L}(\bot) := \bot$, $\mathcal{L}(\top) := \top$, $\mathcal{L}(\alpha) := \alpha$ (for all $\alpha \in TVar$) and $\mathcal{L}(\tau_1 \to \tau_2) := \mathcal{L}(\mu\alpha.\,(\tau_1 \to \tau_2)) := \to$ (for all $\alpha \in TVar$ and $\tau_1, \tau_2 \in$ *can-μTp*).

Figure 2 A normalized version $\mathbf{HB}_{\mathbf{0}}^{=}$ of the coinductive axiomatization for recursive type equality $=_\mu$ given by Brandt and Henglein.

The *axioms* and possible *marked assumptions* in $\mathbf{HB}_{\mathbf{0}}^{=}$:

$$\text{(REFL)}\ \ \overline{\tau = \tau} \qquad\qquad \text{(Assm)}\ \ (\tau = \sigma)^x \quad (\text{with } x \in Mk)\ .$$

The *derivation rules* of $\mathbf{HB}_{\mathbf{0}}^{=}$:

$$\frac{\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma}{\mu\alpha.\,\tau_0 = \sigma}\ \text{FOLD}_l \qquad\qquad \frac{\tau = \sigma_0[\mu\beta.\,\sigma_0/\beta]}{\tau = \mu\beta.\,\sigma_0}\ \text{FOLD}_r$$

$$\frac{\tau = \sigma}{\tau' = \sigma'}\ \text{VAR} \quad \begin{array}{l}(\text{if } \tau' \equiv_{\text{v}} \tau \\ \quad \text{and } \sigma' \equiv_{\text{v}} \sigma)\end{array} \qquad\qquad \frac{\tau_1 = \sigma_1 \qquad \tau_2 = \sigma_2}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}\ \text{ARROW}$$

$$\dfrac{\begin{array}{cc}\langle \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2\rangle^x & \langle \tau_1 \to \tau_2 = \sigma_1 \to \sigma_2\rangle^x \\ \mathcal{D}_1 & \mathcal{D}_2 \\ \tau_1 = \sigma_1 & \tau_2 = \sigma_2\end{array}}{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}\ \begin{array}{l}(\text{ARROW/FIX})_x \\ (\text{if side cond. } \mathbf{I})\end{array}\ .$$

defined by: For all $\tau, \sigma \in \mathit{can}\text{-}\mu Tp$

$$\tau =_\mu \sigma \ :\iff\ \mathbf{Tree}(\tau) = \mathbf{Tree}(\sigma)\ .$$

An example for Definition 2.2 and for the underlying notion of the tree unfolding of a recursive type in $\mathit{can}\text{-}\mu Tp$ is given in Figure 1.

3 The proof systems $\mathbf{HB}_{\mathbf{0}}^{=}$ and $\mathbf{AK}_{\mathbf{0}}^{=}$ for $=_\mu$

In this section we define the two proof systems on which our results will be based: A variant system $\mathbf{HB}_{\mathbf{0}}^{=}$ of the axiomatization for $=_\mu$ given by Brandt and Henglein in [3] and a proof system $\mathbf{AK}_{\mathbf{0}}^{=}$ suitable for consistency-checking similar to a system defined by Ariola and Klop in [2]. We formulate these systems in natural-deduction style and for this and for later purposes we assume a countably infinite set Mk of assumption markers to be given.

Definition 3.1 (The axiom system $\mathbf{HB}_{\mathbf{0}}^{=}$ for $=_\mu$). The formal system $\mathbf{HB}_{\mathbf{0}}^{=}$ has the equations in $\mathit{can}\text{-}\mu Tp\text{–}Eq$ as its *formulas*. It contains the *axioms* (REFL), allows *marked assumptions* (Assm) and has the *derivation rules* VAR, FOLD_l, FOLD_r, ARROW and ARROW/FIX listed in Figure 2. The side condition $\mathbf{I}$ on applications of ARROW/FIX requires that the class of discharged assumptions is actually inhabited, i.e. non-empty.[5] A formula $\tau = \sigma$ is a *theorem* of $\mathbf{HB}_{\mathbf{0}}^{=}$ (symbolically denoted by $\vdash_{\mathbf{HB}_{\mathbf{0}}^{=}} \tau = \sigma$) iff there is a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ with conclusion $\tau = \sigma$, in which all marked assumptions have been discharged at respective applications of ARROW/FIX.

[5] The aim here is to create a clear-cut distinction between applications of ARROW and applications of ARROW/FIX for easing the reasoning about a later defined transformation on proofs.

Figure 3 A normalized 'syntactic-matching' proof system $\mathbf{AK}_{\mathbf{0}}^{=}$ for checking the consistency of given equations with respect to $=_\mu$. This system is related to a one, that was introduced by Ariola and Klop.

The *derivation rules* of $\mathbf{AK}_{\mathbf{0}}^{=}$:

$$\frac{\mu\alpha.\,\tau_0 = \sigma}{\tau_0[\mu\alpha.\,\tau_0/\alpha] = \sigma}\ \text{UNFOLD}_l \qquad \frac{\tau = \mu\beta.\,\sigma_0}{\tau = \sigma_0[\mu\beta.\,\sigma_0/\beta]}\ \text{UNFOLD}_r$$

$$\frac{\tau = \sigma}{\tau' = \sigma'}\ \text{VAR}\quad \begin{array}{l}(\text{if } \tau' \equiv_{\mathrm{v}} \tau \\ \text{and } \sigma' \equiv_{\mathrm{v}} \sigma)\end{array} \qquad \frac{\tau_1 \to \tau_2 = \sigma_1 \to \sigma_2}{\tau_i = \sigma_i}\ \text{DECOMP}\quad (i = 1, 2)\ .$$

Apart from minor differences, the system $\mathbf{HB}_{\mathbf{0}}^{=}$ can be considered as a 'normalized' version of the complete axiomatization for $=_\mu$ given in [3]. No symmetry and transitivity rules are present in $\mathbf{HB}_{\mathbf{0}}^{=}$ and the axioms (FOLD/UNFOLD) used in [3] have been reformulated into the two rules $\text{FOLD}_{l/r}$. $\mathbf{HB}_{\mathbf{0}}^{=}$ is 'normalized' in the sense that it satisfies a version of the *subformula property*. Although lacking the expressivity of symmetry and transitivity rules, the following also holds for $\mathbf{HB}_{\mathbf{0}}^{=}$:

Theorem 3.2 (Soundness and Completeness of $\mathbf{HB}_{\mathbf{0}}^{=}$ with respect to $=_\mu$). *The axiom system $\mathbf{HB}_{\mathbf{0}}^{=}$ is sound and complete with respect to strong recursive type equivalence $=_\mu$, i.e. for all $\tau, \sigma \in$ can-μTp it holds that*

$$\vdash_{\mathbf{HB}_{\mathbf{0}}^{=}} \tau = \sigma \quad \Longleftrightarrow \quad \tau =_\mu \sigma\ .$$

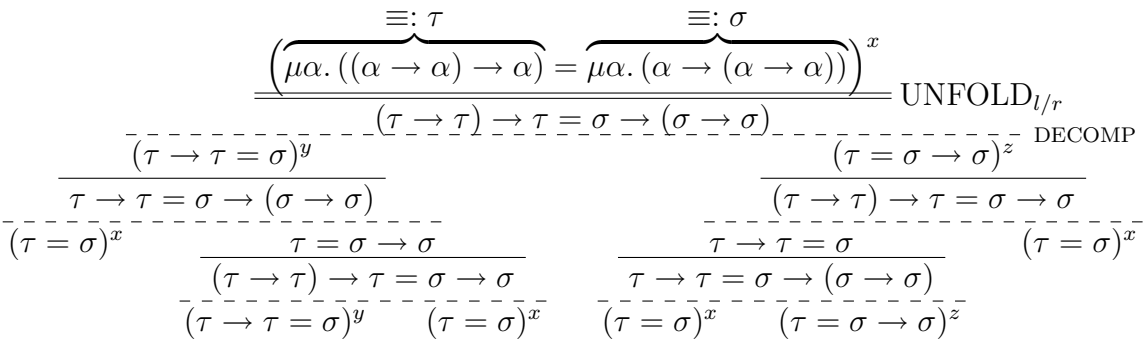
Both the soundness and the completeness of $\mathbf{HB}_{\mathbf{0}}^{=}$ with respect to $=_\mu$ can be shown analogously as done by Brandt and Henglein in [3] for their system. We continue with the definition of a proof system very similar to a 'syntactic-matching' system introduced by Ariola and Klop in Section 3.4 of [2].

Definition 3.3 (A 'syntactic-matching' proof system $\mathbf{AK}_{\mathbf{0}}^{=}$ for $=_\mu$). The formal system $\mathbf{AK}_{\mathbf{0}}^{=}$ contains precisely all equations in *can-μTp–Eq* as its *formulas*. It contains *no axioms*. Its *derivation rules* are the rules VAR, UNFOLD_l, UNFOLD_r and DECOMP that are listed in Figure 3. We will use $\tau = \sigma \vdash_{\mathbf{AK}_{\mathbf{0}}^{=}} \chi_1 = \chi_2$ (for $\tau, \sigma, \chi_1, \chi_2 \in$ *can-μTp*) as notation for the assertion that there is a derivation in $\mathbf{AK}_{\mathbf{0}}^{=}$ from the assumption $\tau = \sigma$ with conclusion $\chi_1 = \chi_2$.

The conspicuous feature of this system is the decomposition rule DECOMP, which is a "destructive" counterpart of the "constructive" composition rules ARROW and ARROW/FIX in $\mathbf{HB}_{\mathbf{0}}^{=}$. Like $\mathbf{HB}_{\mathbf{0}}^{=}$ the system $\mathbf{AK}_{\mathbf{0}}^{=}$ does not contain symmetry and transitivity rules and is 'normalized' in the sense that it fulfills an "inverse subformula property".

Clearly, $\mathbf{AK}_{\mathbf{0}}^{=}$ does not axiomatize $=_\mu$, but a notion of consistency-checking with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$ is sound and complete for $=_\mu$. For being able to state this properly, we need the following terminology: An equation $\tau = \sigma$ between recursive types is a *contradiction with respect to* $=_\mu$ iff $\mathcal{L}(\tau) \neq \mathcal{L}(\sigma)$, i.e. iff the leading symbols of τ and σ differ. Furthermore an equation $\tau = \sigma$ is called $\mathbf{AK}_{\mathbf{0}}^{=}$*-inconsistent* iff there

Figure 4 Assemblage to a finite downwards-growing "tree of consequences" $\mathcal{C}$ of the 6 different possible initial segments of derivations from $\mu\alpha.\,((\alpha \to \alpha) \to \alpha) = \mu\alpha.\,(\alpha \to (\alpha \to \alpha))$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ without VAR-applications until *looping* occurs.



exists a contradiction $\chi_1 = \chi_2$ with respect to $=_\mu$ such that $\tau = \sigma \vdash_{\mathbf{AK}_{\mathbf{0}}^{=}} \chi_1 = \chi_2$; otherwise we say that $\tau = \sigma$ is $\mathbf{AK}_{\mathbf{0}}^{=}$*-consistent*.

Theorem 3.4 (Soundness and Completeness with respect to $=_\mu$ of consistency-checking relative to $\mathbf{AK}_{\mathbf{0}}^{=}$). *Consistency with respect to* $\mathbf{AK}_{\mathbf{0}}^{=}$ *is sound and complete in relation to* $=_\mu$*: For all* $\tau, \sigma \in$ *can-*μTp *it holds that*

$$\tau = \sigma \;\; \textit{is } \mathbf{AK}_{\mathbf{0}}^{=}\textit{-consistent} \quad \Longleftrightarrow \quad \tau =_\mu \sigma \;\; .$$

Sketch of Proof Both the soundness-part "⇒" and the completeness-part "⇐" are easy consequences of the fact that derivations $\mathcal{D}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ from assumption $\tau = \sigma$ with conclusion $\chi_1 = \chi_2$ correspond to computations of $\chi_1, \chi_2 \in$ *can-*μTp with the property that[6] $\mathbf{Tree}(\tau)|p = \mathbf{Tree}(\chi_1)$ and $\mathbf{Tree}(\sigma)|p = \mathbf{Tree}(\chi_2)$ hold for some common position p in the tree unfoldings $\mathbf{Tree}(\tau)$ and $\mathbf{Tree}(\sigma)$ of τ and σ. □

4 The basic observation

It is our next aim to indicate the intuition behind the results of this paper by describing an observation about a concrete and simple example. In this section we keep τ and σ fixed as the two strongly equivalent types in *can-*μTp from Figure 1.

Suppose that we want to prove that the equation $\tau = \sigma$ is indeed consistent with respect to the system $\mathbf{AK}_{\mathbf{0}}^{=}$. Then we are obliged to show for every derivation $\mathcal{D}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ from the assumption $\tau = \sigma$ that the conclusion of $\mathcal{D}$ is not a contradiction with respect to $=_\mu$. But since there are potentially infinitely many such derivations in $\mathbf{AK}_{\mathbf{0}}^{=}$, we might not be able to check all of them in a finite amount of time.

However, it turns out that in every derivation in $\mathbf{AK}_{\mathbf{0}}^{=}$ from $\tau = \sigma$ of depth ≥ 7, that does not contain applications of VAR, a loop arises, i.e. one formula occurs at two different places. What is more, the initial segments until looping occurs of all derivations from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ without VAR-applications can be arranged to the downwards-growing derivation-tree $\mathcal{C}$ depicted in Figure 4. There single and double lines in $\mathcal{C}$ separate the premises and conclusions of applications of UNFOLD$_{l/r}$,

[6] Here $\mathbf{Tree}(\tau)|p$ denotes the subtree of $\mathbf{Tree}(\tau)$ at position p.

Figure 5 The derivation $\mathcal{D}$ of $\mu\alpha.\,((\alpha \to \alpha) \to \alpha) = \mu\alpha.\,(\alpha \to (\alpha \to \alpha))$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumption classes.

$$\dfrac{\dfrac{\dfrac{\dfrac{(\ldots)^{x}}{\tau = \sigma}\quad \dfrac{\dfrac{(\ldots)^{y}}{\tau \to \tau = \sigma}\quad \dfrac{(\ldots)^{x}}{\tau = \sigma}}{(\tau \to \tau) \to \tau = \sigma \to \sigma}}{\tau = \sigma \to \sigma}\,y \qquad \dfrac{\dfrac{(\ldots)^{x}}{\tau = \sigma}\quad \dfrac{(\ldots)^{z}}{\tau = \sigma \to \sigma}}{\tau \to \tau = \sigma \to (\sigma \to \sigma)}}{\tau \to \tau = \sigma \to (\sigma \to \sigma)}}{\tau \to \tau = \sigma}\qquad \dfrac{\dfrac{\dfrac{(\ldots)^{x}}{\tau = \sigma}}{\tau \to \tau = \sigma}\,z}{(\tau \to \tau) \to \tau = \sigma \to \sigma}}{\tau = \sigma \to \sigma}\,(\text{ARR./FIX})_x$$

$$\dfrac{(\tau \to \tau) \to \tau = \sigma \to (\sigma \to \sigma)}{\underbrace{\mu\alpha.\,((\alpha \to \alpha) \to \alpha)}_{\equiv:\tau} = \underbrace{\mu\alpha.\,(\alpha \to (\alpha \to \alpha))}_{\equiv:\sigma}}\,\text{FOLD}_{l/r}$$

whereas branchings at dashed lines stem from the two possible ways in which conclusions can be drawn at rules DECOMP in $\mathbf{AK}_{\mathbf{0}}^{=}$. The markers x, y and z used for some formula occurrences in $\mathcal{C}$ are intended to highlight the looping in those $\mathbf{AK}_{\mathbf{0}}^{=}$-derivations, initial segments of which constitute the branches of $\mathcal{C}$.

It is now possible to use the derivation tree $\mathcal{C}$ in an easy inductive proof[7] for the $\mathbf{AK}_{\mathbf{0}}^{=}$-consistency of the equation $\tau = \sigma$ by combining[8] the following two properties of $\mathcal{C}$: Firstly, as inspection shows, $\mathcal{C}$ does not contain any contradictions with respect to $=_{\mu}$. And secondly, $\mathcal{C}$ can be considered to be the (positive) result of *loop-checking* for all possible derivations without VAR-applications from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$.

In order to give an indication about the particular relationship described in this paper between the systems $\mathbf{AK}_{\mathbf{0}}^{=}$ and $\mathbf{HB}_{\mathbf{0}}^{=}$, we observe[9] the following: By reflecting the downwards-growing derivation tree $\mathcal{C}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ at a horizontal line, it is possible to obtain an upwards-growing prooftree $\text{Refl}(\mathcal{C})$ in the system $\mathbf{HB}_{\mathbf{0}}^{=}$ with occurrences of open assumption classes. Thereby all applications of $\text{UNFOLD}_{l/r}$ in $\mathcal{C}$ are "reflected" into applications of $\text{FOLD}_{l/r}$ in $\text{Refl}(\mathcal{C})$ and all branchings DECOMP into applications of ARROW. To transform $\text{Refl}(\mathcal{C})$ into a derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumptions, it is merely necessary (1) to extend $\text{Refl}(\mathcal{C})$ above each of its leaves by one or two applications of $\text{FOLD}_{l/r}$, (2) to transfer respective assumption markers up to the new formulas at the top of the extended prooftree, and (3) to redirect the bindings described by these markers to respective applications of ARROW below, thereby also changing these into ARROW/FIX-applications. In this way the derivation $\mathcal{D}$ in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumption classes suggestively depicted in Figure 5 is reached.

And similarly, by reflecting the derivation $\mathcal{D}$ from Figure 5 at a horizontal line in an analogous way, it is possible to get a downwards-growing derivation tree $\text{Refl}(\mathcal{D})$ from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ that—although slightly different from $\mathcal{C}$—like $\mathcal{C}$ can be taken

[7] Hereby we mean a proof by induction on the depth $|\mathcal{D}|$ of derivations $\mathcal{D}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ from $\tau = \sigma$.
[8] However, the possible presence of applications of VAR in $\mathbf{AK}_{\mathbf{0}}^{=}$-derivations from the assumption $\tau = \sigma$ does technically complicate a necessary proof by induction to some extent here.
[9] This was noted by J.W. Klop for a similar example in different, but comparable proof systems.

as the basis of an inductive argument for showing the $\mathbf{AK}_{\mathbf{0}}^{=}$-consistency of $\tau = \sigma$.

This example suggests a very direct relationship between derivations in $\mathbf{HB}_{\mathbf{0}}^{=}$ without open assumption classes having conclusion $\tilde{\tau} = \tilde{\sigma}$ (for some $\tilde{\tau}, \tilde{\sigma} \in \mathit{can}\text{-}\mu Tp$) and finite downwards-growing trees of consequences from the same equation $\tilde{\tau} = \tilde{\sigma}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ that are the result of loop-checking and facilitate easy inductive proofs for the consistency of $\tilde{\tau} = \tilde{\sigma}$ relative to $\mathbf{AK}_{\mathbf{0}}^{=}$.

5 The extension e-$\mathbf{HB}_{\mathbf{0}}^{=}$ of $\mathbf{HB}_{\mathbf{0}}^{=}$

For obtaining a precise formulation of the observation in the previous section, it will be helpful to extend the system $\mathbf{HB}_{\mathbf{0}}^{=}$ with three more coinductive fixed-point rules.

Definition 5.1 (The extension e-$\mathbf{HB}_{\mathbf{0}}^{=}$ of the system $\mathbf{HB}_{\mathbf{0}}^{=}$). The extension **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ of the system $\mathbf{HB}_{\mathbf{0}}^{=}$ has the same *formulas* and *axioms* as $\mathbf{HB}_{\mathbf{0}}^{=}$, allows the same *marked assumptions*, and contains all *derivation rules* of $\mathbf{HB}_{\mathbf{0}}^{=}$. Additionally, **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ possesses the rules VAR/FIX, FOLD$_l$/FIX and FOLD$_r$/FIX with applications of the respective form

$$\dfrac{\begin{array}{c}[\,\tau = \sigma\,]^x \\ \mathcal{D}_0 \\ \tau_0 = \sigma_0\end{array}}{\tau = \sigma}\,(R/\mathrm{FIX})_x \qquad \text{(if side cond.(s) } \mathbf{I} \text{ (and } \mathbf{C} \text{ for } R = \mathrm{VAR}))$$

(with some $\tau, \sigma, \tau_0, \sigma_0 \in \mathit{can}\text{-}\mu Tp$ and $x \in Mk$), given that $\dfrac{\tau_0 = \sigma_0}{\tau = \sigma}\,R$ is an application of a rule $R \in \{\,\mathrm{FOLD}_{l/r}, \mathrm{VAR}\,\}$ and that the respectively necessary side conditions described below are satisfied. At such applications the class $[\tau = \sigma]^x$ of open marked assumptions of the form $(\tau = \sigma)^x$ in $\mathcal{D}_0$ gets discharged. The side condition **I** requires that the assumption class $\tau = \sigma$ in $\mathcal{D}_0$ is inhabited (not empty). For applications of VAR/FIX the side condition **C** demands furthermore that $\mathcal{D}_0$ is *contractive* with respect to the marked open assumptions $(\tau = \sigma)^x$, which means that for every thread in $\mathcal{D}_0$ from a marked open assumption $(\tau = \sigma)^x$ downwards at least one application of ARROW or ARROW/FIX is passed. [10]

Although the system **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ is an extension of $\mathbf{HB}_{\mathbf{0}}^{=}$, no new theorems become derivable:

Theorem 5.2 (Equivalence of the systems $\mathbf{HB}_{\mathbf{0}}^{=}$ and e-$\mathbf{HB}_{\mathbf{0}}^{=}$). *The system* **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ *is a conservative extension of* $\mathbf{HB}_{\mathbf{0}}^{=}$ *and hence*[11] *the systems* $\mathbf{HB}_{\mathbf{0}}^{=}$ *and* **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ *are equivalent (i.e. they possess the same theorems). More specifically, every derivation* $\mathcal{D}$ *in* **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ *can effectively be transformed into a derivation* $\mathcal{D}'$ *in* $\mathbf{HB}_{\mathbf{0}}^{=}$ *with the same conclusion and the same (if any) open assumption classes.*

[10] It is easy to see that either of two following more special requirements $\mathbf{C}_1$ and $\mathbf{C}_2$ could have been used instead of the side condition **C** for applications of VAR/FIX with an equivalent definition as the result: $\mathbf{C}_1$ is the condition "$\mathcal{D}_0$ contains at least one application of ARROW or ARROW/FIX" and $\mathbf{C}_2$ demands that "there is at least one application of a rule different from VAR in $\mathcal{D}_0$".

[11] Since $\mathbf{HB}_{\mathbf{0}}^{=}$ and **e-**$\mathbf{HB}_{\mathbf{0}}^{=}$ have the same formulas.

6 Consistency-Unfoldings

In a second step of the formulation of the observation in Section 4 into a precise statement, we will formalize finite downwards-growing trees of consequences in $\mathbf{AK}_{\mathbf{0}}^{=}$ as "consistency-unfoldings", which allow to prove easily the $\mathbf{AK}_{\mathbf{0}}^{=}$-consistency of the formulas at their respective roots. – We have to give a definition of "partial consistency-unfoldings" first.

Definition 6.1 (Partial Consistency-Unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$). For all recursive types $\tau, \sigma \in \mathit{can}\text{-}\mu Tp$ a *partial consistency-unfolding* (a *p.c.u.*) $\mathcal{C}$ of the equation $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ is a finite downwards-growing "tree of consequences" of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ that together with the assertion "$\mathcal{C}$ is a p.c.u. of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$" can be formed by a finite number of applications of the following 5 generation rules. Thereby the notion of an *unbound leaf-occurrence of a marked formula* (an *u.l.o.m.f.*) in a p.c.u. is defined in parallel: [12]

(i) For all $\tau, \sigma \in \mathit{can}\text{-}\mu Tp$ and $x \in Mk$ $\boxed{(\tau = \sigma)^x}$ is a p.c.u. $\mathcal{C}$ from $\tau = \sigma$. The occurrence of $(\tau = \sigma)^x$ in $\mathcal{C}$ is the single u.l.o.m.f. in $\mathcal{C}$. – Furthermore for all $\tau \in \mathit{can}\text{-}\mu Tp$ $\boxed{\tau = \tau}$ is a p.c.u. of $\tau = \tau$, which contains no u.l.o.m.f.'s in $\mathcal{C}$.

(ii) For all $\tau, \sigma, \tau_0, \sigma_0 \in \mathit{can}\text{-}\mu Tp$ $\boxed{\dfrac{\tau = \sigma}{(\tau_0 = \sigma_0)^{\boldsymbol{m}_0}}\, R \quad \mathcal{C}_0}$ is a p.c.u. $\mathcal{C}$ of $\tau = \sigma$ given that $\mathcal{C}_0$ is a p.c.u. of $\tau_0 = \sigma_0$ and that R is an application of a rule $\mathrm{UNFOLD}_{l/r}$ or VAR. An u.l.o.m.f. in $\mathcal{C}$ is such an occurrence of a marked formula in $\mathcal{C}$ within its subtree $\mathcal{C}_0$ that corresponds to an u.l.o.m.f. in $\mathcal{C}_0$.

(iii) For all $\tau, \sigma, \tau_0, \sigma_0 \in \mathit{can}\text{-}\mu Tp$ and $x \in Mk$ $\boxed{\dfrac{(\tau = \sigma)^x}{(\tau_0 = \sigma_0)^{\boldsymbol{m}_0}}\, R \quad \mathcal{C}_0 \quad [\tau = \sigma]^x}$ is a p.c.u. $\mathcal{C}$ of $\tau = \sigma$ given that (1) $\mathcal{C}_0$ is a p.c.u. of $\tau_0 = \sigma_0$ in which the (indicated) class $[\tau = \sigma]^x$ of all u.l.o.m.f.'s of the form $(\tau = \sigma)^x$ is non-empty and that either (2a) R is an application of a rule $\mathrm{UNFOLD}_{l/r}$ or (2b) R is an application of VAR and $\mathcal{C}_0$ contains at least one application of a rule different from VAR. All occurrences of $(\tau = \sigma)^x$ within the subtree $\mathcal{C}_0$ of $\mathcal{C}$, that correspond to u.l.o.m.f.'s in $\mathcal{C}_0$, are *bound back* in $\mathcal{C}$ to the occurrence of $(\tau = \sigma)^x$ at the root. For all marked formulas $(\tilde{\tau} = \tilde{\sigma})^{\tilde{x}}$ different from $(\tau = \sigma)^x$ the unbound leaf-occurrences of this marked formula correspond uniquely and in an obvious

[12] In the following clauses the addition "in $\mathbf{AK}_{\mathbf{0}}^{=}$" in statements like "$\mathcal{C}$ is a p.c.u. in $\mathbf{AK}_{\mathbf{0}}^{=}$" is always dropped. Auxiliary framed boxes are used to delimit the defined p.c.u.'s from the surrounding text. Here and later we will allow formulas $(\tau = \sigma)^{\boldsymbol{m}}$ with $\tau, \sigma \in \mathit{can}\text{-}\mu Tp$ and a boldface-marker $\boldsymbol{m}$ to stand either (a) for the unmarked formula $\tau = \sigma$ or (b) for a marked formula $(\tau = \sigma)^x$ with some marker $x \in Mk$, which is furthermore assumed to be denoted by $\boldsymbol{m}$ in this case.

way to the u.l.o.'s of $(\tilde{\tau} = \tilde{\sigma})^{\tilde{x}}$ within the subtree $\mathcal{C}_0$ of $\mathcal{C}$.

(iv) $\dfrac{\begin{array}{c}\tau_{01} \to \tau_{02} = \sigma_{01} \to \sigma_{02}\end{array}}{\begin{array}{cc}(\tau_{01} = \sigma_{01})^{\boldsymbol{m}_{01}} & (\tau_{02} = \sigma_{02})^{\boldsymbol{m}_{02}} \\ \mathcal{C}_{01} & \mathcal{C}_{02}\end{array}}\ \text{DECOMP}$ is a p.c.u. $\mathcal{C}$ of the formula $\tau_{01} \to \tau_{02} = \sigma_{01} \to \sigma_{02}$ for all $\tau_{01}, \tau_{02}, \sigma_{01}, \sigma_{02} \in \textit{can-}\mu Tp$, given that $\mathcal{C}_{0i}$ is a p.c.u. of $\tau_{0i} = \sigma_{0i}$ for each $i \in \{1, 2\}$. The u.l.o.m.f.'s in $\mathcal{C}$ correspond uniquely and in an obvious way to the u.l.o.m.f.'s in either of its immediate subtrees $\mathcal{C}_{01}$ or $\mathcal{C}_{02}$.

(v) $\dfrac{\begin{array}{c}(\tau_{01} \to \tau_{02} = \sigma_{01} \to \sigma_{02})^{x}\end{array}}{\begin{array}{cc}(\tau_{01} = \sigma_{01})^{\boldsymbol{m}_{01}} & (\tau_{02} = \sigma_{02})^{\boldsymbol{m}_{02}} \\ \mathcal{C}_{01} & \mathcal{C}_{02} \\ \langle\tau = \sigma\rangle^{x} & \langle\tau = \sigma\rangle^{x}\end{array}}\ \text{DECOMP}$ (with some $x \in Mk$ and with $\tau :\equiv \tau_{01} \to \tau_{02}$ and $\sigma :\equiv \sigma_{01} \to \sigma_{02}$) is a p.c.u. $\mathcal{C}$ of $\tau_{01} \to \tau_{02} = \sigma_{01} \to \sigma_{02}$ for all $\tau_{01}, \tau_{02}, \sigma_{01}, \sigma_{02} \in \textit{can-}\mu Tp$ given that $\mathcal{C}_{0i}$ is a p.c.u. of $\tau_{0i} = \sigma_{0i}$ for each $i \in \{1, 2\}$ and that there is at least one unbound leaf-occurrence of the marked formula $(\tau_{01} \to \tau_{02} = \sigma_{01} \to \sigma_{02})^{x}$ in either $\mathcal{C}_{01}$ or in $\mathcal{C}_{02}$. All occurrences of $(\tau = \sigma)^{x}$ within either of the immediate subtrees $\mathcal{C}_{01}$ and $\mathcal{C}_{02}$ of $\mathcal{C}$, that correspond to u.l.o.m.f.'s in $\mathcal{C}_{01}$ or $\mathcal{C}_{02}$, are *bound back* in $\mathcal{C}$ to the occurrence of $(\tau = \sigma)^{x}$ at the root (and hence are not u.l.o.m.f.'s in $\mathcal{C}$). For every marked formula $(\tilde{\tau} = \tilde{\sigma})^{\tilde{x}}$ different from $(\tau = \sigma)^{x}$ the unbound leaf-occurrences of this marked formula correspond uniquely and in an obvious way to the u.l.o.'s of $(\tilde{\tau} = \tilde{\sigma})^{\tilde{x}}$ within either of the sub-p.c.u.'s $\mathcal{C}_{01}$ or $\mathcal{C}_{02}$ of $\mathcal{C}$.

The *depth* $|\mathcal{C}|$ of a p.c.u. $\mathcal{C}$ is defined as the depth of the underlying (derivation-) tree.

Definition 6.2 (Consistency-Unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$). Let τ and σ be recursive types in canonical form. A partial consistency-unfolding $\mathcal{C}$ of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ is called a *consistency-unfolding* (a *c.u.*) of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ if and only if $\mathcal{C}$ does not contain any unbound leaf-occurrences of marked formulas.

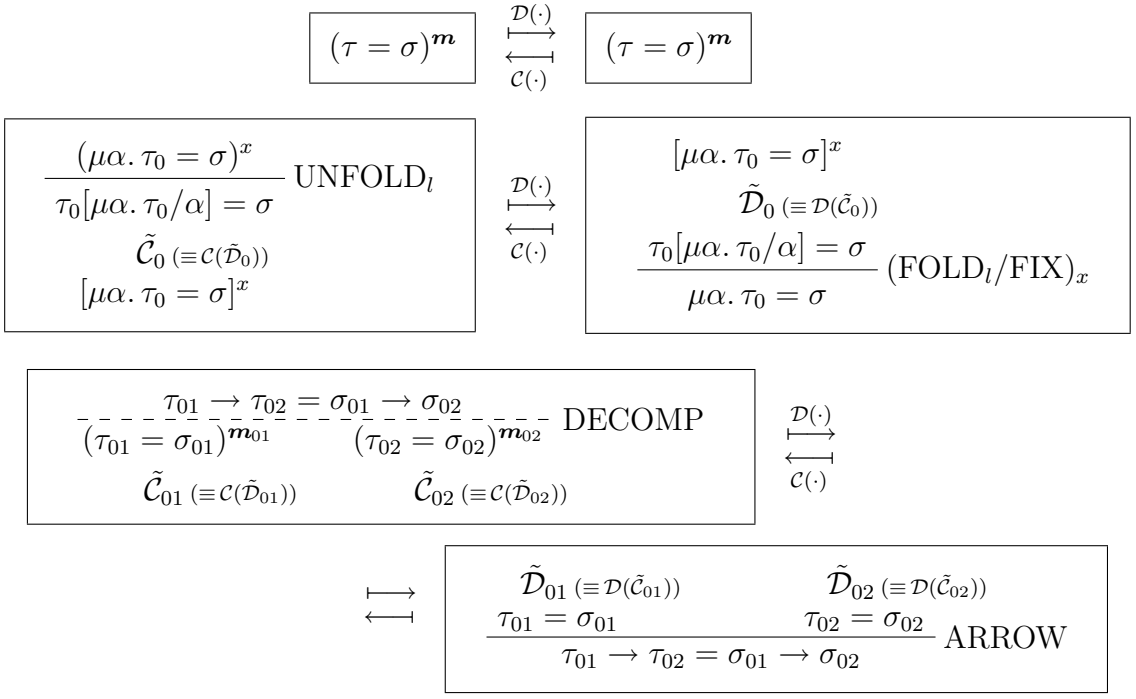
According to these definitions the derivation tree $\mathcal{C}$ depicted in Figure 4 can now be recognized as a p.c.u. in $\mathbf{AK}_{\mathbf{0}}^{=}$ without u.l.o.m.f.'s and hence as a consistency-unfolding of $\mu\alpha.\,((\alpha \to \alpha) \to \alpha) = \mu\alpha.\,(\alpha \to (\alpha \to \alpha))$ in $\mathbf{AK}_{\mathbf{0}}^{=}$. – The following theorem establishes the link motivated by the example in Section 4 between the notions of "$\mathbf{AK}_{\mathbf{0}}^{=}$-consistency" and "consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{=}$".

Theorem 6.3 *For all recursive types $\tau, \sigma \in \textit{can-}\mu Tp$ it holds that:*

$$\tau = \sigma \textit{ is } \mathbf{AK}_{\mathbf{0}}^{=}\textit{-consistent} \iff \textit{There exists a consistency-unfolding of } \tau = \sigma \textit{ in } \mathbf{AK}_{\mathbf{0}}^{=}\ . \tag{6.1}$$

Hint at the Proof Let $\tau, \sigma \in \textit{can-}\mu Tp$. The implication "$\Leftarrow$" in (6.1) follows by induction on the depth of an arbitrary derivation from $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$, in which

Figure 6 Three exemplary cases of inductive clauses in the definitions of the reflection mappings $\mathcal{D}(\cdot)$ and $\mathcal{C}(\cdot)$ between partial consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ and derivations in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ (with possibly open assumption classes).



induction it is used that every given c.u. of $\tau = \sigma$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ combines *in some sense* all initial segments of such derivations until looping occurs. The implication "$\Rightarrow$" in (6.1) follows by an analogous, in fact as good as 'dual', argument to that one used in a proof (following [3]) for the completeness of $\mathbf{HB}_{\mathbf{0}}^{=}$ with respect to $=_\mu$. □

7 A duality between the proof systems $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ and $\mathbf{AK}_{\mathbf{0}}^{=}$

In a third step of our formalization of the observation in Section 4, we give a definition of a pair of reflection mappings $\mathcal{D}(\cdot)$ and $\mathcal{C}(\cdot)$ between p.c.u.'s in $\mathbf{AK}_{\mathbf{0}}^{=}$ and derivations in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$.

Definition 7.1 (Reflection Mappings $\mathcal{D}(\cdot)$ and $\mathcal{C}(\cdot)$). The *reflection mapping* $\mathcal{D}(\cdot)$ from partial consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{=}$ to derivations in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ (with possibly open assumption classes) is defined by induction on the depth $|\tilde{\mathcal{C}}|$ of a p.c.u. $\tilde{\mathcal{C}}$ according to 5 inductive clauses, which refer to the 5 cases in the inductive definition of p.c.u.'s in Definition 6.1. Three exemplary cases (one each referring to items (i) and (iv) and one regarding the subcase for an UNFOLD$_l$-rule of item (iii) in Definition 6.1) have been depicted in Figure 6. The definition of the *reflection mapping* $\mathcal{C}(\cdot)$ in the opposite direction can be carried out for all derivations $\tilde{\mathcal{D}}$ in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ by induction on its depth $|\tilde{\mathcal{D}}|$ with clauses that apart from the base case distinguish the 8 cases of different rules in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$, applications of which may occur as the last rule applications in $\tilde{\mathcal{D}}$; three of the nine clauses are depicted in Figure 6.

The well-definedness of $\mathcal{D}(\cdot)$ and $\mathcal{C}(\cdot)$ as functions between the set of p.c.u.'s in $\mathbf{AK}_{\mathbf{0}}^{=}$ and the set of derivations in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ with possibly open assumption classes can be shown by induction on the depth of the elements in the domain of the respective mapping. – We are now able to state our main theorem.

Theorem 7.2 (A Duality between derivations in e-HB$_{\mathbf{0}}^{=}$ and consistency-unfoldings in AK$_{\mathbf{0}}^{=}$). *There is a bijective functional relationship between derivations in* $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ *without open assumption classes and consistency-unfoldings in* $\mathbf{AK}_{\mathbf{0}}^{=}$ *via the reflection mappings* $\mathcal{D}(\cdot)$ *and* $\mathcal{C}(\cdot)$ *defined in Definition 7.1 in the following sense:*

(i) For every consistency-unfolding $\tilde{\mathcal{C}}$ *of* $\tau = \sigma$ *in* $\mathbf{AK}_{\mathbf{0}}^{=}$ *(with some* $\tau, \sigma \in$ *can-*μTp*) its reflection* $\mathcal{D}(\tilde{\mathcal{C}})$ *is a derivation in* $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ *with conclusion* $\tau = \sigma$ *and without open assumption classes.*

(ii) For every derivation $\tilde{\mathcal{D}}$ *in* $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ *without open assumption classes and with conclusion* $\tau = \sigma$ *(for some* $\tau, \sigma \in$ *can-*μTp*) its reflection* $\mathcal{C}(\tilde{\mathcal{D}})$ *is a consistency-unfolding of* $\tau = \sigma$ *in* $\mathbf{AK}_{\mathbf{0}}^{=}$*.*

(iii) The functions $\mathcal{D}(\cdot)$ *of taking the reflection of a consistency-unfolding in* $\mathbf{AK}_{\mathbf{0}}^{=}$ *and* $\mathcal{C}(\cdot)$ *of taking the reflection of a derivation in* $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ *without open assumption-classes are each other's inverse.*

The very immediate kind of this bijective functional relationship and the possibility to visualize the reflection functions in a geometrical way is reason to call it a duality*.*

Sketch of Proof All three items of the theorem (the third one can be split into the two assertions $\mathcal{D} \circ \mathcal{C} = \mathrm{id}$ and $\mathcal{C} \circ \mathcal{D} = \mathrm{id}$) can be shown by quite straightforward inductions using the inductive clauses in the definitions of $\mathcal{D}(\cdot)$ and $\mathcal{C}(\cdot)$. In these inductions bookkeeping must be done for respectively the set of open marked assumptions in an $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$-derivation or the classes of u.l.o.m.f.'s in a p.c.u. in $\mathbf{AK}_{\mathbf{0}}^{=}$. □

An example of a pair $(\tilde{\mathcal{D}}, \tilde{\mathcal{C}})$ consisting of a derivation $\tilde{\mathcal{D}}$ in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ and a consistency-unfolding $\tilde{\mathcal{C}}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ that are each other's reflection via the operations $\mathcal{D}(\cdot)$ and $\mathcal{C}(\cdot)$ is depicted in Figure 7.

8 A duality in proof systems for bisimulation equivalence on cyclic term graphs

In this section we want to sketch how our duality result about two proof systems for recursive type equality can be transferred to similar proof systems concerned with bisimulation equivalence on equational representations of cyclic term graphs.

In the aim to limit technicalities and to follow [2], we will only consider equational specifications of cyclic term graphs without free variables. We are assuming a countably infinite set *RVar* of *recursion variables* to underlie the following definition (we will let small Greek letters $\alpha, \beta, \ldots$ vary through recursion variables).

Definition 8.1 (Canonical Term Graph Specifications). Let Σ be a first-order signature. A *canonical term graph specification* (a *c.t.g.s.*) is an equational

Figure 7 Example consisting of a consistency-unfolding $\tilde{\mathcal{C}}$ in $\mathbf{AK}_{\mathbf{0}}^{=}$ and of a derivation $\tilde{\mathcal{D}}$ in $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ that are each other's *reflection* via the mappings $\mathcal{D}(\cdot)$ and $\mathcal{C}(\cdot)$: This means that $\mathcal{D}(\tilde{\mathcal{C}}) = \tilde{\mathcal{D}}$ and $\mathcal{C}(\tilde{\mathcal{D}}) = \tilde{\mathcal{C}}$ hold for $\tilde{\mathcal{C}}$ and $\tilde{\mathcal{D}}$ as considered below.

$$\tilde{\mathcal{D}} := \left\{ \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\tau = \sigma)^x \qquad \overline{\bot = \bot}}{\tau \to \bot = \sigma \to \bot}\,\text{ARROW}}{\tau = \sigma \to \bot}}{\tau \to \bot = (\sigma \to \bot) \to \bot}\,\text{FOLD}_r \qquad \overline{\bot = \bot}}{\tau \to \bot = \sigma}\,\text{ARROW}}{\underbrace{\mu\alpha.\,(\alpha \to \bot)}_{\equiv:\,\tau} = \underbrace{\mu\beta.\,((\beta \to \bot) \to \bot)}_{\equiv:\,\sigma}}\,(\text{FOLD}_l/\text{FIX})_x \right.$$

$$\tilde{\mathcal{C}} := \left\{ \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\mu\alpha.\,(\alpha \to \bot) = \mu\beta.\,((\beta \to \bot) \to \bot))^x}{\tau \to \bot = \sigma}\,\text{UNFOLD}_l}{\tau \to \bot = (\sigma \to \bot) \to \bot}\,\text{UNFOLD}_r \qquad \overline{\bot = \bot}}{\tau = \sigma \to \bot}\,\text{DECOMP}}{\tau \to \bot = \sigma \to \bot}\,\text{DECOMP}}{(\tau = \sigma)^x \qquad \bot = \bot} \right.$$

specification of the form $\langle\, \alpha_0 \,|\, \{\alpha_0 = t_0, \ldots, \alpha_n = t_n\} \,\rangle$, where $n \in \mathbb{N}$, $\alpha_0, \ldots, \alpha_n$ pairwisely different recursion variables in $RVar$ and for all i with $0 \leq i \leq n$ the terms t_i are of the form $t_i \equiv F(\alpha_{i1}, \ldots, \alpha_{in_i})$ for some function symbol $F \in \Sigma$ of arity n_i and variables $\alpha_{i1}, \ldots, \alpha_{in_i} \in \{\alpha_0, \ldots, \alpha_n\}$. We will use the letters g and h to vary through c.t.g.s.'s and denote by $\mathcal{TGS}(\Sigma)$ the set of all c.t.g.s.'s over Σ.

Bisimilarity between c.t.g.s.'s is defined in [2] as follows:

Definition 8.2 (Bisimulation Equivalence $\leftrightarrows$ on c.t.g.s.'s). Let Σ be a signature. Let g and h be canonical term graph specifications over Σ of the form $g = \langle\, \alpha_0 \,|\, \{\alpha_0 = t_0, \ldots, \alpha_n = t_n\} \,\rangle$ and $h = \langle\, \alpha'_0 \,|\, \{\alpha'_0 = t'_0, \ldots, \alpha'_{n'} = t'_{n'}\} \,\rangle$.

(a) R is called a *bisimulation* between g and h if and only if
   (i) R is a relation with domain $\{\alpha_0, \ldots, \alpha_n\}$ and codomain $\{\alpha'_0, \ldots, \alpha'_{n'}\}$;
   (ii) $\alpha_0 \,R\, \alpha'_0$;
   (iii) if $\alpha_i \,R\, \alpha'_j$ for some i, j with $0 \leq i \leq n$ and $0 \leq j \leq n'$, and given that $t_i \equiv F(\alpha_{i1}, \ldots, \alpha_{in_i})$ and $t'_j \equiv F'(\alpha'_{j1}, \ldots, \alpha'_{jn'_j})$ with some $n_i, n'_j \in \mathbb{N}_0$, then $F \equiv F'$ (and hence $n_i = n'_j$) and $\alpha_{i1} \,R\, \alpha'_{j1}$, ... , $\alpha_{in_i} \,R\, \alpha'_{jn'_j}$ must hold.

(b) We say that g and h are *bisimilar* (symbolically denoted by $g \leftrightarrows h$) iff there exists a bisimulation between g and h.

Example 8.3 We consider the two canonical term graph specifications

$$g := \langle\, \alpha_0 \,|\, E_g \,\rangle := \langle\, \alpha_0 \,|\, \{\alpha_0 = F(\alpha_1, \alpha_2),\ \alpha_1 = F(\alpha_0, \alpha_2),\ \alpha_2 = G(\alpha_1, \alpha_0)\} \,\rangle \quad (8.1)$$
$$h := \langle\, \beta_0 \,|\, E_h \,\rangle := \langle\, \beta_0 \,|\, \{\beta_0 = F(\beta_0, \beta_1),\ \beta_1 = G(\beta_0, \beta_0)\} \,\rangle \quad (8.2)$$

Figure 8 A Brandt–Henglein-like axiomatization $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$ without symmetry and transitivity rules of bisimulation equivalence between canonical term graph specifications over signature Σ.

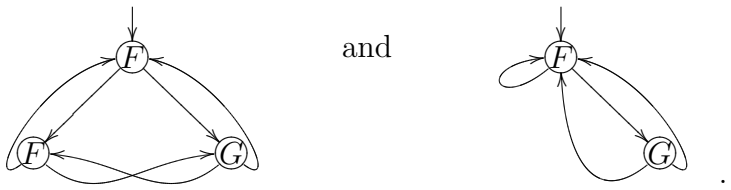
The *axioms* and possible *marked assumptions* in $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$:

$$\text{(REFL)}\ \ \overline{\langle\, \alpha \,|\, \{\alpha = C,\, \ldots\}\,\rangle = \langle\, \beta \,|\, \{\beta = C,\, \ldots\}\,\rangle}\ \ (\text{if } C \text{ is a constant symbol in } \Sigma) \qquad \text{(Assm)}\ \ (g = h)^x\ \ (\text{with } x \in Mk)\ .$$

The *derivation rules* of $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$: Rules COMP and rules COMP/FIX with

$$\frac{\begin{array}{cccc} \{\!\{\, \langle\, \alpha \,|\, E_g \,\rangle = \langle\, \beta \,|\, E_h \,\rangle \,\}\!\}^x & & \{\!\{\, \langle\, \alpha \,|\, E_g \,\rangle = \langle\, \beta \,|\, E_h \,\rangle \,\}\!\}^x & \\ \mathcal{D}_1 & & \mathcal{D}_n & \\ \langle\, \alpha_1 \,|\, E_g \,\rangle = \langle\, \beta_1 \,|\, E_h \,\rangle & \ldots & \langle\, \alpha_n \,|\, E_g \,\rangle = \langle\, \beta_n \,|\, E_h \,\rangle & \end{array}}{\langle\, \alpha \,|\, \underbrace{\{\alpha = F(\alpha_1, \ldots, \alpha_n)\} \uplus E_g^{(0)}}_{= E_g} \,\rangle = \langle\, \beta \,|\, \underbrace{\{\beta = F(\beta_1, \ldots, \beta_n)\} \uplus E_h^{(0)}}_{= E_h} \,\rangle}\ (\text{COMP/FIX})_x\ (\text{if s.-c. } \mathbf{I})$$

in $\mathcal{TGS}(\{F, G\})$. These correspond respectively to the two cyclic term graphs



.

It is easy to check that $R := \{(\alpha_0, \beta_0), (\alpha_1, \beta_0), (\alpha_2, \beta_1)\}$ is a bisimulation between g and h according to Definition 8.2. Hence $g \leftrightarrows h$ holds, i.e. g and h are bisimilar.

A sound and complete axiom system $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$ for $\leftrightarrows$, which is very similar to the 'normalized' version $\mathbf{HB}_{\mathbf{0}}^{=}$ of the axiom system for $=_\mu$ by Brandt and Henglein, is depicted in Figure 8. Just as for its counterpart in $\mathbf{HB}_{\mathbf{0}}^{=}$, the rule ARROW/FIX, applications of the rule COMP/FIX in $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$ are subjected to the side condition **I**: This requires that the discharged assumption class is in fact non-empty (to distinguish such applications from ones of the "plain" COMP-rule). Soundness and completeness of $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$ with respect to $\leftrightarrows$ means, that

$$\vdash_{\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}} g = h \quad \Longleftrightarrow \quad g \leftrightarrows h$$

holds for all $g, h \in \mathcal{TGS}(\Sigma)$ (for some signature Σ).

A 'syntactic matching' proof system $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ for $\leftrightarrows$, which is similar to the system $\mathbf{AK}_{\mathbf{0}}^{=}$, is depicted in Figure 9, its single derivation rule being the decomposition rule DECOMP. $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ is sound and complete with respect to $\leftrightarrows$ in the sense that

$$g = h \ \text{ is } \mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}\text{-consistent} \quad \Longleftrightarrow \quad g \leftrightarrows h$$

holds for $g, h \in \mathcal{TGS}(\Sigma)$ (for some signature Σ). Hereby an equation $\tilde{g} = \tilde{h}$ between two c.t.g.s.'s $\tilde{g}$ and $\tilde{h}$ is called $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$-*consistent* iff no contradiction with respect to $\leftrightarrows$ is derivable in $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ from $\tilde{g} = \tilde{h}$. And furthermore an equation $\tilde{g} = \tilde{h}$ be-

Figure 9 A 'syntactic-matching' proof system $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ for testing bisimulation equivalence on equations between canonical term graph specifications.

The *derivation rules* of $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$:

$$\frac{\langle\, \alpha \,|\, \overbrace{\{\alpha = F(\alpha_1,\ldots,\alpha_n)\} \uplus E_g^{(0)}}^{=E_g} \,\rangle = \langle\, \beta \,|\, \overbrace{\{\beta = F(\beta_1,\ldots,\beta_n)\} \uplus E_h^{(0)}}^{=E_h} \,\rangle}{\langle\, \alpha_i \,|\, E_g \,\rangle = \langle\, \beta_i \,|\, E_h \,\rangle}\ \text{DECOMP}\ (\text{for } 1 \leq i \leq n)$$

tween two c.t.g.s.'s $\tilde{g} = \langle\, \alpha_0 \,|\, \{\alpha_0 = t_0, \ldots\} \,\rangle$ and $\tilde{h} = \langle\, \alpha'_0 \,|\, \{\alpha'_0 = t'_0, \ldots\} \,\rangle$ is agreed to be a *contradiction with respect to* $\leftrightarrows$ iff it holds that $t_0 \equiv F(\alpha_{01}, \ldots, \alpha_{0n_0})$ and $t'_0 \equiv G(\alpha'_{01}, \ldots, \alpha'_{0n'_0})$ for some $n_0, n'_0 \in \mathbb{N}_0$, variables $\alpha_{01}, \ldots, \alpha_{0n_0}$, $\alpha'_{01}, \ldots, \alpha'_{0n'_0}$ and *different* symbols $F, G \in \Sigma$ (i.e. $F \not\equiv G$).

Now it is very straightforward to define the notion of p.c.u.'s and consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ of equations between c.t.g.s.'s analogously to Definitions 6.1 and 6.2. And furthermore also reflection functions $\mathcal{C}(\cdot)$ and $\mathcal{D}(\cdot)$ between p.c.u.'s in $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ and derivations in $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$ can be defined very similar to (and in fact easier than in) Definition 7.1. In this way we are lead to the following counterpart of Theorem 7.2 for the two proof systems considered here.

Theorem 8.4 (A Duality between derivations in $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$ and consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$). *There is a bijective functional relationship between derivations in $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$ without open assumption classes and consistency-unfoldings in $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ via reflection mappings $\mathcal{C}(\cdot)$ and $\mathcal{D}(\cdot)$: This means that completely analogous statements to that in items (i), (ii) and (iii) of Theorem 7.2 are true.*

In Figure 10 the assertion of this theorem is exemplified for the c.t.g.s.'s g and h of Example 8.3 by a suggestively typeset pair $(\tilde{\mathcal{D}}, \tilde{\mathcal{C}})$ of a derivation $\tilde{\mathcal{D}}$ for $g = h$ in $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrows}$ without open assumption classes and a consistency-unfolding of $g = h$ in $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ that are each other's "mirror image" via reflection mappings $\mathcal{C}(\cdot)$ and $\mathcal{D}(\cdot)$.

9 Conclusion

In the main part of this paper we have motivated and developed a precise formal relationship between two different proof systems concerned with recursive type equality $=_\mu$ on (a very small class of) recursive types. We showed the existence of a bijective correspondence, which can geometrically be visualized, between (1) derivations in an extension $\mathbf{e\text{-}HB}_{\mathbf{0}}^{=}$ of a normalized version $\mathbf{HB}_{\mathbf{0}}^{=}$ of the axiomatization for $=_\mu$ by Brandt and Henglein and (2) what we defined as "consistency-unfoldings" in a proof system $\mathbf{AK}_{\mathbf{0}}^{=}$ à la Ariola and Klop for equational testing with respect to $=_\mu$. This correspondence takes place via two reflection mappings that formalize effective transformations and that are inverse to each other.

In the last section we indicated that the described duality result is not specific to the two considered proof systems for recursive types: We sketched an analogous duality theorem for a similar pair of proof systems concerned with the notion of

Figure 10 Example consisting of a derivation in $\mathbf{HB}_{\mathbf{0}}^{\leftrightarrow}$ without open assumption classes and of a consistency-unfolding in $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrow}$ that are each other's "reflection". (The canonical term graph specifications g and h are taken from Example 8.3).

$$\dfrac{\dfrac{(\ldots)^x \qquad \dfrac{(\langle\, \alpha_1 \,|\, E_g \,\rangle = \langle\, \beta_0 \,|\, E_h \,\rangle)^y \qquad (\ldots)^x}{\langle\, \alpha_2 \,|\, E_g \,\rangle = \langle\, \beta_1 \,|\, E_h \,\rangle}\,\mathrm{y}}{\langle\, \alpha_1 \,|\, E_g \,\rangle = \langle\, \beta_0 \,|\, E_h \,\rangle} \qquad \dfrac{\dfrac{(\ldots)^x \qquad (\ldots)^z}{\langle\, \alpha_1 \,|\, E_g \,\rangle = \langle\, \beta_0 \,|\, E_h \,\rangle} \qquad (\ldots)^x}{\langle\, \alpha_2 \,|\, E_g \,\rangle = \langle\, \beta_1 \,|\, E_h \,\rangle}\,\mathrm{z}}{\underbrace{\langle\, \alpha_0 \,|\, E_g \,\rangle}_{=g \text{ in } (8.1)} = \underbrace{\langle\, \beta_0 \,|\, E_h \,\rangle}_{=h \text{ in } (8.2)}}\,\mathrm{x}$$

$$\dfrac{\dfrac{(\langle\, \alpha_0 \,|\, E_g \,\rangle = \langle\, \beta_0 \,|\, E_h \,\rangle)^x}{(\langle\, \alpha_1 \,|\, E_g \,\rangle = \langle\, \beta_0 \,|\, E_h \,\rangle)^y} \qquad \dfrac{(\langle\, \alpha_0 \,|\, E_g \,\rangle = \langle\, \beta_0 \,|\, E_h \,\rangle)^x}{(\langle\, \alpha_2 \,|\, E_g \,\rangle = \langle\, \beta_1 \,|\, E_h \,\rangle)^z}}{(\ldots)^x \qquad \dfrac{\langle\, \alpha_2 \,|\, E_g \,\rangle = \langle\, \beta_1 \,|\, E_h \,\rangle}{(\langle\, \alpha_1 \,|\, E_g \,\rangle = \langle\, \beta_0 \,|\, E_h \,\rangle)^y \qquad (\ldots)^x} \qquad \dfrac{\langle\, \alpha_1 \,|\, E_g \,\rangle = \langle\, \beta_0 \,|\, E_h \,\rangle}{(\ldots)^x \qquad (\ldots)^z} \qquad (\ldots)^x}$$

bisimulation equivalence on equational specifications of cyclic term graphs.

Apart from establishing a precise formal link between the systems $\mathbf{HB}_{\mathbf{0}}^{=}$ and $\mathbf{AK}_{\mathbf{0}}^{=}$ by tying together closely the notions of "derivability in $\mathbf{HB}_{\mathbf{0}}^{=}$" and "consistency with respect to $\mathbf{AK}_{\mathbf{0}}^{=}$", the main significance of our duality result Theorem 7.2 consists perhaps in the following: It can be used to understand and justify the soundness of the—at least at first sight—seemingly paradoxical reasoning formalized by the rule ARROW/FIX in $\mathbf{HB}_{\mathbf{0}}^{=}$. In fact, our results facilitate an alternative soundness proof for the system $\mathbf{HB}_{\mathbf{0}}^{=}$, which is independent from the one given in [3], by 'reducing' the soundness of $\mathbf{HB}_{\mathbf{0}}^{=}$ to the soundness of the system $\mathbf{AK}_{\mathbf{0}}^{=}$.

A slightly more detailed version of this paper is available on the web via the link `http://www.cs.vu.nl/~clemens/termgraph2002_ext.ps` . Forthcoming work [4] is concerned with a detailed study of proof-theoretic transformations between the mentioned proof systems for recursive types and a number of variant systems.

References

[1] Amadio, R.M., Cardelli, L.: "Subtyping Recursive Types", *ACM Transactions on Programming Languages and Systems 15 (4)*, pp. 575–631, 1993.

[2] Ariola, Z.M., Klop, J.W.: "Equational Term Graph Rewriting", *Fundamenta Informaticae 26 (3,4)*, pp. 207–240, June 1996; extended version as: *Vrije Universiteit Amsterdam Technical Report IR–391*, September 1995.

[3] Brandt, M., Henglein, F.: "Coinductive Axiomatization of Recursive Type Equality and Subtyping", *Fundamenta Informaticae 33*, pp. 1–30, 1998.

[4] Grabmayer, C.: "Proof-Theoretic Interconnections between Proof Systems for Recursive Type Equality", forthcoming as *Vrije Universiteit Amsterdam Technical Report*, 2002.