Math. Struct. in Comp. Science (2007), *vol.* 17, *pp.* 439–484.
doi:10.1017/S0960129507006111 Printed in the United Kingdom

A duality between proof systems for cyclic term graphs

CLEMENS GRABMAYER[†]

Department of Computer Science, Vrije Universiteit Amsterdam, de Boelelaan 1081a, 1081 HV Amsterdam, the Netherlands
Email: clemens@cs.vu.nl.

Received 1 September 2005; revised 4 September 2006

This paper presents a proof-theoretic observation about two kinds of proof systems for bisimilarity between cyclic term graphs.

First we consider proof systems for demonstrating that μ-term specifications of cyclic term graphs have the same tree unwinding. We establish a close connection between adaptations for μ-terms over a general first-order signature of the coinductive axiomatisation of recursive type equivalence by Brandt and Henglein (Brandt and Henglein 1998) and of a proof system by Ariola and Klop (Ariola and Klop 1995) for consistency checking. We show that there exists a simple duality by mirroring between derivations in the former system and formalised consistency checks, which are called 'consistency unfoldings', in the latter. This result sheds additional light on the axiomatisation of Brandt and Henglein: it provides an alternative soundness proof for the adaptation considered here.

We then outline an analogous duality result that holds for a pair of similar proof systems for proving that equational specifications of cyclic term graphs are bisimilar.

1. Introduction

Proof systems for regular cyclic objects have a long tradition in both logic and computer science. Well-known examples are the axiomatisations of regular expression equivalence by Salomaa (Salomaa 1966), of bisimilarity between regular behaviours by Milner (Milner 1984), of the equivalence and subtyping relations between recursive types by Amadio and Cardelli (Amadio and Cardelli 1993) and of bisimilarity between cyclic term graphs by Ariola and Klop (Ariola and Klop 1995). These systems are essentially algebraic in character, with the rules of equational logic, including composition (congruence), being part of their logical apparatus, and with induction being an important tool for proving soundness and completeness.

More recently, proof systems with a coalgebraic background have been formulated in which proofs are able to employ circular reasoning, exploiting the fact that many

[†] This paper was written while the author was employed on the NWO-project *GeoProc* – 'Geometry of Processes', Nr. 612.000.313. (NWO=Nederlandse Organisatie voor Wetenschappelijk Onderzoek.)

equivalence problems allow reformulations in terms of the existence of finite or finitely representable bisimulations. in addition to their coinductive foundations, such systems were first introduced by Brandt and Henglein for recursive type equivalence and subtyping (Brandt and Henglein 1998). Hüttel and Stirling had earlier given a system with a similar form for bisimilarity of normed recursive processes over the process algebra BPA but without an explicit coinductive motivation (Hüttel and Stirling 1991). These systems, and a number of similar ones, have in common the presence of inference rules with applications in which assumptions of the form of the conclusion may be discharged. Such inferences allow us, roughly speaking, to detect that a bisimulation-building process that is formalised by a derivation has reached a subtask that it has already solved before. Completeness of such coalgebraic systems can often be shown by proving appropriate coinduction principles and by linking derivations with bisimulations.

Ariola and Klop presented a third kind of proof system for bisimilarity between cyclic term graphs (Ariola and Klop 1995). Here the focus is on consistency with the system: an equation between equational specifications of two cyclic term graphs can be added consistently if and only if the term graphs are bisimilar. The most prominent feature of this system is a decomposition rule, which allows us to compare the inner structure of the terms on either side of an equation. An application of this rule on the premise of a term equation $F(s_1,\ldots,s_n) = F(t_1,\ldots,t_n)$ (where F is an n-ary function symbol) syntactically matches, for $i \in \{1,\ldots,n\}$, a subterm s_i of the term on the left-hand side of the equation with the corresponding subterm t_i of the term on the right-hand side, that is, it allows us to infer $s_i = t_i$. This feature of the decomposition rule has led Ariola and Klop to call their proof system a 'syntactic-matching' system.

Proving that an equation is consistent with a syntactic-matching system amounts to showing that, assuming the equation, it is not possible to derive a 'contradiction', an equation between terms that have different leading symbols. However, because there are usually infinitely many possible derivations from a given equation, it is, in general, not possible to decide the consistency of a given equation by a naive search procedure that successively generates all possible derivations and checks their conclusions for contradictions. But in the case of syntactic-matching systems for *regular* cyclic objects, which by successive decompositions give rise to only a finite number of reachable objects, decidability of consistency with the system can often be shown. This is because in such a situation it is easier to analyse the termination behaviour of a consistency-deciding procedure that first generates a systematic overview of all possible derivations from a given equation until looping occurs, and then: if a contradiction is generated in the course of an execution, we have shown the inconsistency of the equation; if the systematic overview is completed and no contradiction is found, we have proved its consistency.

Such a loop-check procedure for derivations in a syntactic-matching system can actually be linked to the detection of a finite bisimulation with a circular structure between the objects (for example terms) on either side of a given equation. Procedures of a similar kind are used in inference rules and in concrete implementations of the concept of 'circular coinduction' that has been introduced and developed in a sequence of papers by Goguen and Rosu, starting with Rosu and Goguen (2001).

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

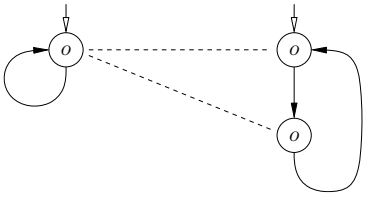

Figure 1. The bisimilar term graphs $\mathscr{G}_1$ and $\mathscr{H}_1$, where the label o is a unary function symbol.

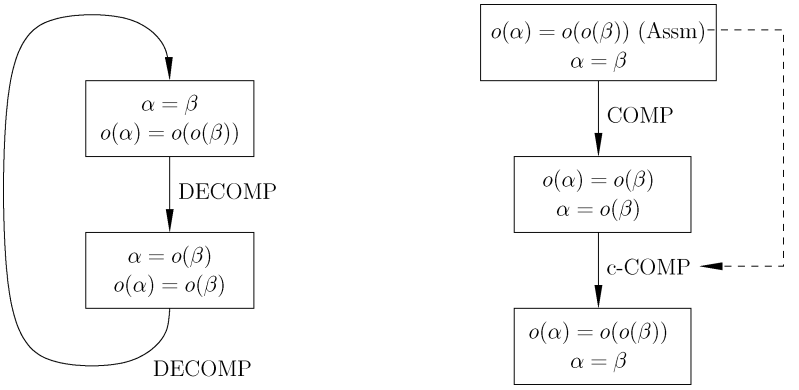

Figure 2. A looping consistency-check in the system of Ariola and Klop (on the left) and a closely related proof in the system of Brandt and Henglein (on the right).

Klop noted (Klop 2000) that there exists, on the syntactic level, a conspicuous similarity between:

(a) trying to find a derivation for an equation in the system of Brandt and Henglein; and
(b) trying to demonstrate the consistency of the same equation in a related syntactic-matching system by using a loop-check procedure.

This observation was the starting point for a detailed investigation into the proof-theoretic connection between the Brandt–Henglein and Ariola–Klop (syntactic-matching) systems, which led to the results presented in this paper. As a motivation, we can illustrate the relationship mentioned above using a simple example.

Let $\mathscr{G}_1$ and $\mathscr{H}_1$ be the term graphs shown in Figure 1, which can be represented by the equational specifications $g_1 = \langle\, \alpha \,|\, \alpha = o(\alpha) \,\rangle$ and $h_1 = \langle\, \beta \,|\, \beta = o(o(\beta)) \,\rangle$, respectively. $\mathscr{G}_1$ and $\mathscr{H}_1$ are bisimilar in an intuitive way: a bisimulation is specified by the broken lines in Figure 1, adhering to the definition of bisimilarity for term graphs due to Ariola and Klop (Ariola and Klop 1995). (Here o is a unary function symbol and α and β are recursion variables.) Moreover, g_1 and h_1 possess the same tree unwinding: the infinite unary tree in which every node is labelled by the unary function symbol o.

We now look at a proof, illustrated on the left-hand side in Figure 2, showing that g_1 and h_1 are bisimilar by means of a looping consistency check in a syntactic-matching

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

system. This proof presupposes two things:

1 a syntactic-matching system in which the only inference rules are decomposition, and expanding variables in equational specifications according to their definition at the outermost position; and
2 a correspondence result of bisimilarity between term graphs with consistency relative to the syntactic-matching system.

An explanation of this graphical proof is given below.

Assuming that g_1 and h_1 are bisimilar leads us to adopt the equation $\alpha = \beta$ between the leading variables α and β of the term graph specifications g_1 and h_1 (see this equation in the top box on the left of Figure 2). By applying the rule for expanding the definitions for α and β in g_1 and h_1 at the outermost positions, we can derive the equations $o(\alpha) = \beta$, $\alpha = o(o(\beta))$ and $o(\alpha) = o(o(\beta))$. Since $o(\alpha) = o(o(\beta))$ is the only formula the decomposition rule (the second rule in the system) can be applied to, it is the only one we have included at the bottom of the topmost box, as it is the only relevant formula here (a box contains formulas derivable by inferences that do not change the represented term graphs). From this formula, an application of the decomposition rule DECOMP strips off the leftmost occurrences of the function symbol o, resulting in the equation $\alpha = o(\beta)$ in the box below. Expanding the definition of α in g_1 then leads to $o(\alpha) = o(\beta)$. Finally, a second application of DECOMP gives us back the equation $\alpha = \beta$ from which we started as an assumption. In this way we have shown that all long enough derivations from $\alpha = \beta$ in the syntactic-matching system are circular and loop back to this same equation. Since we do not encounter any 'contradictions' (that is, no equations between terms with different leading symbols) during any of these derivations, the equation $\alpha = \beta$ is consistent with the system.

Hence, the 'deduction graph' on the left-hand side of Figure 1 can be viewed as a looping consistency-check that witnesses the consistency of the equation $\alpha = \beta$ with the syntactic-matching system. It follows, by the presupposed correspondence result, that the represented term graphs $\mathscr{G}_1$ and $\mathscr{H}_1$ are bisimilar. It is also noticeable that we can quite easily extract the bisimulation indicated in Figure 1 from the cycling derivations.

Now it turns out that the looping consistency-check on the left-hand side of Figure 1 is closely related to a derivation in a Brandt–Henglein proof system: by mirroring the 'deduction graph' on the left in a horizontal line, a cyclic deduction graph with bottommost formula $\alpha = \beta$ is reached. By subsequently turning the arrows around, a cyclic derivation is obtained in a proof system that contains two rules: a composition rule COMP and a rule for shortening equations by applying the definitions of the recursion variables in equational specifications at the outermost positions. At this point we observe that in a Brandt–Henglein system we can get a circular composition rule c-COMP with an application of the form

$$\dfrac{\begin{array}{c}[\,o(\alpha) = o(o(\alpha))\,]^u \\ \mathscr{D}_1 \\ \alpha = o(\alpha)\end{array}}{o(\alpha) = o(o(\alpha))}\ \text{c-COMP},\ u \tag{1.1}$$

(here u acts as an assumption marker, indicating that the set of yet undischarged assumptions $(o(\alpha) = o(o(\alpha)))^u$ carrying this marker from the top is discharged at the

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

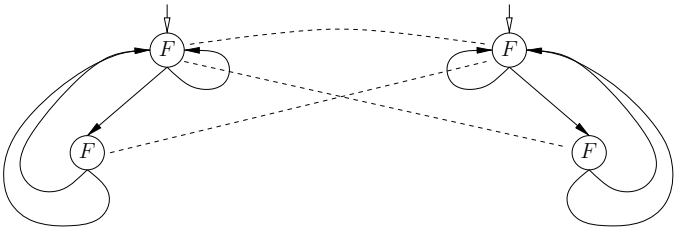

Figure 3. The bisimilar term graphs $\mathscr{G}_2$ and $\mathscr{H}_2$, where the label F is a binary function symbol.

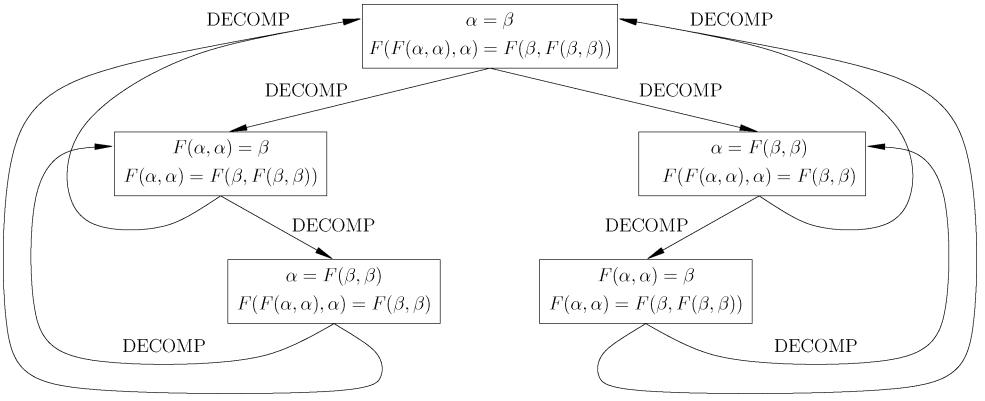

Figure 4. Determining bisimilarity of the term graphs represented as $\langle\, \alpha \mid \alpha = F(F(\alpha,\alpha),\alpha) \,\rangle$ and as $\langle\, \beta \mid \beta = F(\beta,F(\beta,\beta)) \,\rangle$ by a loop-check procedure using decomposition steps.

application of c-COMP at the bottom). Assumptions of the form of the conclusion may be discharged at such an application. By using an application of the form (1.1), we obtain the Brandt–Henglein derivation on the right-hand side of Figure 2: we remove the outgoing deduction from the bottom box, but place a second copy of the bottom box on the top, from which the box in the middle is reachable by a COMP application, and we replace the COMP application between the lower two boxes by an application of the circular decomposition rule c-COMP at which the assumption $o(\alpha) = o(o(\alpha))$ is discharged.

As a second example, consider the bisimilar term graphs $\mathscr{G}_2$ and $\mathscr{H}_2$ in Figure 3, which can be represented by the equational specifications $g_2 = \langle\, \alpha \mid \alpha = F(F(\alpha,\alpha),\alpha) \,\rangle$ and $h_2 = \langle\, \beta \mid \beta = F(\beta,F(\beta,\beta)) \,\rangle$. A looping consistency check for this pair of graphs is shown in Figure 4. We can again obtain a derivation in a Brandt–Henglein proof system in this case by mirroring the consistency check in a horizontal line upwards, by changing applications of decomposition rules into applications of composition rules, by adding a set of additional boxes and by discharging the assumptions arising at circular decomposition rules below. (Such a Brandt–Henglein derivation, but formalised for μ-term representations of the term graphs $\mathscr{G}_2$ and $\mathscr{H}_2$, will be encountered later in Figure 12.)

In this paper, starting from the observation described above, we extract a proof-theoretic result that links the Ariola–Klop and Brandt–Henglein systems: there exists a duality via mirroring between 'consistency unfoldings' (formalisations of successful consistency

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

checks) in the Ariola–Klop system and derivations in the Brandt–Henglein system. While we are pincipally concerned with proof systems for μ-term representations of cyclic term graphs (Sections 3–5 and 7), we also outline an analogous result for proof systems concerned with equational specifications of cyclic term graphs (Section 7.5, using Section 6).

The relevance of our results

It is well known that there is a duality between algebra and coalgebra in the sense that statements in an algebraic setting have counterparts in a coalgebraic setting, and *vice versa*, by merely turning the arrows around in statements in the framework of category theory (and in illustrative diagrams). However, we feel that it is not well known that this duality in category theory and the connection with bisimilarity find an expression in the fact that Brandt–Henglein proofs are the exact mirror image of proofs in a bisimulation-based proof search using decomposition (Ariola–Klop proofs).

In particular, we want to emphasise the following four aspects of the duality results proved here:

1 They can explain the *a priori* counter-intuitive, circular Brandt–Henglein proofs in terms of the more intuitive notion of bisimilarity.
2 They reveal an intimate link between the soundness and completeness of the Brandt and Henglein and Ariola and Klop systems.
3 They exhibit the duality between composition and decomposition rules, thus corroborating the fundamental perception between constructing and deconstructing operations, and between algebra and coalgebra.
4 They capture an algebra/coalgebra duality in a different way from the turning around of arrows in the step between algebraic to coalgebraic formulations in category theory.

Previous work

A first account of duality results as reported in this article was given in a paper presented at the TERMGRAPH 2002 Workshop, Barcelona, October 7, 2002 (Grabmayer 2002a). There, and in a superseding technical report (Grabmayer 2002b), the duality is formulated for a pair of proof systems for recursive type equivalence, establishing a direct link from the coinductive axiomatisation for this relation by Brandt and Henglein to a syntactic-matching system. In the context of an investigation devoted exclusively to proof-theoretic connections between proof systems for recursive type equivalence, duality statements are presented in Chapter 6 of the thesis Grabmayer (2005).

Overview of this paper

Basic definitions for μ-terms over a general signature and canonical specifications of cyclic term graphs are gathered together in Section 2.

In Section 3, we define an adaptation of tree unwinding equivalence on μ-terms for the coinductive axiomatisation due to Brandt and Henglein, and we formulate a soundness and completeness theorem for this system.

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

In Section 4, we give an adaptation of tree unwinding equivalence on μ-terms for the syntactic-matching system due to Ariola and Klop. We define a concept of consistency with respect to this system, and formulate a correspondence theorem that links consistency with respect to the system to tree unwinding equivalence.

In Section 5, we introduce variants of these two proof systems that have stronger proof-theoretic properties, but facilitate the same completeness and correspondence results. We also define an extension of the variant Brandt–Henglein system with more coinductive rules.

In Section 6, we define a pair of similar proof systems concerned with bisimilarity on equational specifications of cyclic term graphs.

The duality results are developed in Section 7. We introduce the concept of 'consistency unfolding' in the variant Ariola–Klop system, and define 'mirroring functions' that map consistency unfoldings into derivations in the extended variant Brandt–Henglein system, and *vice versa*. Following these preparations, we prove that the mirroring functions define a bijective correspondence between consistency unfoldings in the variant Ariola–Klop system and derivations in the extended variant Brandt–Henglein system. As an application, we give an alternative soundness proof of the variant Brandt–Henglein system. We outline an analogous duality result linking the proof systems from Section 6 that is concerned with bisimilarity on equational specifications of cyclic term graphs.

Finally, in Section 8, we report on generalisations of our results and put a number of questions for further research.

2. Preliminaries

In this section we introduce some basic definitions concerning μ-term representations (in Section 2.1) and canonical term graph specifications of cyclic term graphs (in Section 2.2), but we begin by fixing some global notation.

We use ω to mean the set $\{0, 1, 2, 3, \ldots\}$. For a finite set A, an *alphabet*, we use A^* to denote the set of (finite) words over A. The empty word is designated by ϵ. Concatenation of words w and w' is denoted multiplicatively as $w.w'$. Let A and B be sets. We use $f : A \to B$ to denote a (total) function, and $f : A \rightharpoonup B$ to denote a partial function between A and B. For a partial function $f : A \rightharpoonup B$, we use $f(a)\!\downarrow$ and $f(a)\!\uparrow$ to abbreviate the statements 'f is defined for a' and 'f is undefined for a', respectively, and use $\mathrm{dom}(f) =_{\mathrm{def}} \{\, x \in A \mid f(x)\!\downarrow \}$ to denote the *domain* of f.

A *signature* Σ is a non-empty set of *function symbols* that is equipped with a function $\mathsf{arity} : \Sigma \to \omega$ that to every function symbol $F \in \Sigma$ assigns its *arity*. Function symbols with arity zero are called *constants*. We will use capital letters F, G, H for function symbols, and C, D for constants. For a signature Σ and all $n \in \omega$, we use Σ_n to denote the subset of Σ containing all function symbols of arity n.

2.1. *Preliminaries for μ-terms*

In this section we gather together the basic definitions for μ-terms over a given signature, which will be used in the proof systems investigated in Sections 3–5 and 7. We define

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

the set of μ-terms over a given signature, the concepts of the 'leading symbol' and 'tree unwinding' of a μ-term, and the 'tree unwinding equivalence' relation. We define a concept of bisimulation between μ-terms based on rewrite relations for μ-terms such as 'unfolding' and 'decomposition', and relate it to tree unwinding equivalence by means of a 'finite bisimulation principle' for proving that μ-terms have the same tree unwinding.

We begin with the definition of the set of μ-terms over a given signature.

Definition 2.1. Let *Var* be a countably infinite set of *variables* and Σ be a signature. The set $Ter_\mu(\Sigma)$ of *μ-terms over Σ (and Var)* is defined inductively by the following three clauses:

(i) For all variables $\alpha \in Var$, $\alpha \in Ter_\mu(\Sigma)$.
(ii) For all n-ary function symbols F in Σ, if $t_1, \ldots, t_n \in Ter_\mu(\Sigma)$, then $F(t_1, \ldots, t_n) \in Ter_\mu(\Sigma)$ also.
(iii) For all $\alpha \in Var$, if $t \in Ter_\mu(\Sigma)$, then $\mu\alpha.\, t \in Ter_\mu(\Sigma)$ also.

For all $s, t \in Ter_\mu(\Sigma)$, we say that s is a *subterm* of t (notation $s \trianglelefteq t$) if and only if s is t or if s precedes t in the formation of t according to the definition above.

The set *Var* of variables will always be treated as given implicitly and its presence underlying this definition is therefore not reflected in the notation $Ter_\mu(\Sigma)$. We use small Greek letters α, β, γ (which may be indexed, primed, and so on) as syntactical variables for variables, and letters t, s, r (which may be indexed, primed, and so on) as syntactical variables for μ-terms.

μ is a binder: occurrences of α in t within $\mu\alpha.\, t$ are bound. An occurrence of a variable α in a μ-term t that is not in the scope of a μ-binder is called *free*; otherwise it is called *bound*. We adopt the practice of working modulo renaming of bound variables (that is, we work implicitly with renaming equivalence classes of μ-terms). We use the symbol $\equiv$ to denote syntactic equality of μ-terms up to the renaming of bound variables. We use $\mu\alpha_1\alpha_2 \ldots \alpha_n.\, t$ as an abbreviated notation for μ-terms like $\mu\alpha_1.\, \mu\alpha_2. \ldots \mu\alpha_n.\, t$.

For all $\alpha \in Var$ and $s, t \in Ter_\mu(\Sigma)$, a *substitution expression* $t[s/\alpha]$ denotes the result of substituting s for all free occurrences of α in t. Some care has to be taken while performing substitutions in order to avoid the capture of free variables: bound variables will need to be renamed sometimes before carrying out a substitution.

Let $[\,]$ be a constant, called a *context-hole*, such that $[\,] \notin \Sigma$. A *context C over Σ* is a μ-term in $Ter_\mu(\Sigma \cup \{[\,]\})$ with precisely one occurrence of $[\,]$ in it. Contexts are not considered modulo renaming. For a context C over Σ, and a μ-term $t \in Ter_\mu(\Sigma)$, we use $C[t]$ to denote the result of replacing $[\,]$ in C by t (free variables in t may get bound).

The signature Σ will be considered as a parameter for all definitions and statements given here, and it will be carried along in the notation, as, for example, for the set $Ter_\mu(\Sigma)$ of μ-terms over Σ. Except when a signature Σ is specified locally (as some concrete set or as obeying some particular conditions), definitions and statements should be understood to apply for all possible choices of signature Σ.

Making use of the fact that μ-bindings can be used to describe back-pointers in cyclic term graphs, μ-terms can be viewed as term specifications of cyclic term graphs. An explicit translation $\mathscr{G}$ of μ-terms into cyclic term graphs was described in Ariola and

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

Klop (1995). For example, the term graphs $\mathscr{G}_1$ and $\mathscr{H}_1$ in Figure 1 are the images under $\mathscr{G}$ of the μ-terms $\mu\alpha.\, o(\alpha)$ and $\mu\beta.\, o(o(\beta))$, respectively; and the term graphs $\mathscr{G}_2$ and $\mathscr{H}_2$ in Figure 3 are the images under $\mathscr{G}$ of the μ-terms $\mu\alpha.\, F(F(\alpha,\alpha),\alpha)$ and $\mu\beta.\, F(\beta,F(\beta,\beta))$. There is the notable fact, shown by Blom, that the image of the translation $\mathscr{G}$ can be characterised as the class of cyclic term graphs without 'horizontal sharing' (Blom 2001). The translation $\mathscr{G}$, however, is not a total function because there are μ-terms such as $\mu\alpha.\, \alpha$ and $\mu\alpha\beta\gamma.\, \beta$ that do not correspond to cyclic term graphs.

Relying on the translation $\mathscr{G}$ and on the well-known concept of 'tree unwinding' for cyclic term graphs, it is possible to assign to each μ-term its 'tree unwinding', which is a potentially infinite labelled term tree. Although the 'tree unwinding' concept for μ-terms can also be introduced formally in this way, we give a more direct definition here for two reasons:

1 We want to base the definition on a syntactic characterisation of those μ-terms that do not represent cyclic term graphs (that is, on which the translation $\mathscr{G}$ is undefined).
2 The specific definition of the 'tree unwinding' of a μ-term given below will be needed later in the proof of the 'finite bisimulation principle'.

In preparation for the definition of 'tree unwinding', we define the 'leading symbol' of a μ-term t, which is intended to denote the symbol that labels the root of the tree unwinding of t (this relationship is stated formally in Proposition 2.4). The leading symbol function will be left undefined for μ-terms like $\mu\alpha.\, \alpha$ and $\mu\alpha\beta\gamma.\, \beta$, which do not represent cyclic term graphs under the translation $\mathscr{G}$.

The *leading symbol (partial) function* $\mathscr{L} : Ter_\mu(\Sigma) \rightharpoonup \Sigma \cup Var$ is defined by induction on the structure of μ-terms over Σ by the clauses

$$\begin{aligned} \mathscr{L}(\alpha) &=_{\text{def}} \alpha \\ \mathscr{L}(F(t_1,\ldots,t_n)) &=_{\text{def}} F \\ \mathscr{L}(\mu\alpha.\, t) &=_{\text{def}} \begin{cases} \uparrow & \ldots\ \mathscr{L}(t)\uparrow \text{ or } \mathscr{L}(t) = \alpha \\ \mathscr{L}(t) & \ldots\ \mathscr{L}(t)\downarrow \text{ and } \mathscr{L}(t) \neq \alpha \end{cases} \end{aligned}$$

(for all $\alpha \in Var$, $n \in \omega$, $F \in \Sigma_n$, and $t, t_1, \ldots, t_n \in Ter_\mu(\Sigma)$). If $\mathscr{L}(t)\downarrow$, for $t \in Ter_\mu(\Sigma)$, then $\mathscr{L}(t)$ is called the *leading symbol* of t and we say that t *has a defined leading symbol*; otherwise we say that the leading symbol of t is undefined. For example, the leading symbol of $\mu\alpha\beta.\, F(\alpha,\beta)$ is the binary function symbol F, and the leading symbol of both of the μ-terms $\mu\alpha.\, \alpha$ and $\mu\alpha\beta\gamma.\, \beta$ is undefined.

The set of μ-terms in $Ter_\mu(\Sigma)$ with undefined leading symbol can be characterised by the following proposition, which is not difficult to prove.

Proposition 2.2. For all $t \in Ter_\mu(\Sigma)$,

$$\begin{aligned} \mathscr{L}(t)\uparrow \iff\ & (\exists\, n \in \omega)\,(\exists\, \alpha_0, \alpha_1, \ldots, \alpha_n \in Var) \\ & (\exists\, j \in \{0,1,\ldots,n\})\,\big[\, t \equiv \mu\alpha_0\alpha_1 \ldots \alpha_n.\, \alpha_j \,\big]\,. \end{aligned}$$

Remark 2.3. Note that μ-terms with undefined leading symbol are related, via a well-known translation of μ-terms into the λ-calculus, to unsolvable λ-terms, that is, λ-terms

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

that do not possess a head normal form. Assuming a chosen fixed-point combinator Y, this translation interprets μ-bindings '$\mu(\cdot)$' by '$Y\,\lambda(\cdot)$', that is, a λ-binding that is given to Y as an application. More precisely, a translation $\mathscr{T}$ from $Ter_\mu(\Sigma)$ to λ-terms with function symbols from Σ is inductively defined by the clauses

$$\begin{aligned} \mathscr{T}(\alpha) &=_{\text{def}} \alpha \\ \mathscr{T}(F(t_1,\ldots,t_n)) &=_{\text{def}} F(\mathscr{T}(t_1),\ldots,\mathscr{T}(t_n)) \\ \mathscr{T}(\mu\alpha.\,t) &=_{\text{def}} Y\,\lambda\alpha.\,\mathscr{T}(t) \end{aligned}$$

(for all $\alpha \in Var$, $n \in \omega$, $F \in \Sigma_n$, and $t, t_1, \ldots, t_n \in Ter_\mu(\Sigma)$), where Y is a fixed-point combinator. It is not difficult to prove for this translation that for all $t \in Ter_\mu(\Sigma)$, the leading symbol of t is undefined if and only if $\mathscr{T}(t)$ is unsolvable, that is, if $\mathscr{T}(t)$ does not possess a head normal form.

Since we are interested in μ-terms as specifications of cyclic term graphs, from now on we will not consider μ-terms t for which the leading symbol of t or of one of its subterms is undefined. In other words, we restrict our attention to the set

$$T_\mu(\Sigma) =_{\text{def}} \{t \in Ter_\mu(\Sigma) \mid (\forall s \in Ter_\mu(\Sigma),\, s \trianglelefteq t)\,[\,\mathscr{L}(s)\!\downarrow]\}$$

of all those μ-terms t in $Ter_\mu(\Sigma)$ for which each subterm has a defined leading symbol. By definition, the set $T_\mu(\Sigma)$ is closed under the subterm relation. It can be shown that $T_\mu(\Sigma)$ is precisely the subset of $Ter_\mu(\Sigma)$ consisting of all those μ-terms over Σ for which the translation $\mathscr{G}$ to cyclic term graphs is defined.

As a prerequisite for the definition of the 'tree unwinding' of a μ-term, we need a formalisation of, possibly infinite, term trees. For this, we define a *term tree* over Σ to be a partial function $T : \omega^* \rightharpoonup \Sigma \cup Var$ with the property that the domain of T, the set $Acc(T) =_{\text{def}} \text{dom}(T)$ of *access paths* or *nodes* of T, fulfills two properties:

(i) $Acc(T)$ is non-empty, and prefix-closed.

(ii) The arity of the symbol $T(p)$ that labels a node p in T determines the number of successors of p in T (variables are assumed to have zero arity).

We use $\mathbb{T}(\Sigma)$ to denote the *set of all term trees* over Σ.

For all $T \in \mathbb{T}(\Sigma)$ and for all $p \in Acc(t)$, the *subtree* $T|p$ of T *determined by* p is the partial function $T|_p : \omega^* \rightharpoonup \Sigma \cup Var$ defined by $T|_p(\tilde{p}) =_{\text{def}} T(p.\tilde{p})$ for all $\tilde{p} \in \omega^*$. It is easy to verify that subtrees of term trees are again term trees.

Now we define the 'tree unwinding' of a μ-term over Σ. The function

$$\begin{aligned} \boldsymbol{T} \;:\; T_\mu(\Sigma) \longrightarrow \mathbb{T}(\Sigma \cup Var)\,, \quad t \;\longmapsto\; \boldsymbol{T}(t) : \omega^* &\rightharpoonup \Sigma \cup Var \\ p \;&\mapsto\; \boldsymbol{T}(t)(p) \end{aligned}$$

that assigns to every $t \in T_\mu(\Sigma)$ its *tree unwinding* $\boldsymbol{T}(t)$ is defined by induction on the length $|p|$ of access paths $p \in \omega^*$, with a subinduction on the number of leading μ-bindings in t,

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

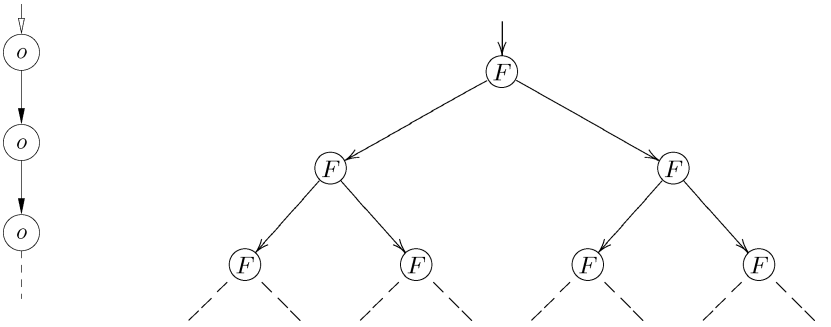

Figure 5. On the left: the tree unwinding of the μ-terms $s_1 \equiv \mu\alpha.\, o(\alpha)$ and $t_1 \equiv \mu\beta.\, o(o(\beta))$, which specify the cyclic term graphs $\mathscr{G}_1$ and $\mathscr{H}_1$ in Figure 1, respectively. On the right: the common tree unwinding of the μ-terms $s_2 \equiv \mu\alpha.\, F(F(\alpha,\alpha),\alpha)$ and $t_2 \equiv \mu\beta.\, F(\beta,F(\beta,\beta))$, which specify the cyclic term graphs $\mathscr{G}_2$ and $\mathscr{H}_2$ in Figure 3, respectively.

by the following clauses:

$$\begin{aligned}
\boldsymbol{T}(\alpha)(\epsilon) &=_{\text{def}} \alpha \\
\boldsymbol{T}(\alpha)(i.p) &\uparrow \\
\boldsymbol{T}(F(t_1,\ldots,t_n))(\epsilon) &=_{\text{def}} F \\
\boldsymbol{T}(F(t_1,\ldots,t_n))(i.p) &=_{\text{def}} \begin{cases} \boldsymbol{T}(t_{i+1})(p) & \ldots\ i < n \\ \uparrow & \ldots\ i \geqslant n \end{cases} \\
\boldsymbol{T}(\mu\alpha.\, t)(p) &=_{\text{def}} \boldsymbol{T}(t[\mu\alpha.\, t/\alpha])(p)
\end{aligned}$$

(for all $\alpha \in \mathit{Var}$, $n \in \omega$, $F \in \Sigma_n$, $t, t_1, \ldots, t_n \in T_\mu(\Sigma)$, and for all $i \in \omega$ and $p \in \omega^*$).

Figure 5 contains an example where the common tree unwindings of μ-terms that specify the bisimilar cyclic term graphs in Figures 1 and 3, respectively, are shown informally.

As stated by the following easily verifiable proposition, the symbol that labels the root of the tree unwinding of a μ-term t in $T_\mu(\Sigma)$ is just the leading symbol of t.

Proposition 2.4. For all $t \in T_\mu(\Sigma)$, we have $\boldsymbol{T}(t)(\epsilon) = \mathscr{L}(t)$.

Using the concept of the tree unwinding of a μ-term, we now define the binary equivalence relation $=_{\boldsymbol{T}}$ on $T_\mu(\Sigma)$ as the property of 'having the same tree unwinding'.

Definition 2.5. The binary relation $=_{\boldsymbol{T}}$ on $T_\mu(\Sigma)$, which is called *tree unwinding equivalence*, is defined, for all $s, t \in T_\mu(\Sigma)$, by

$$s =_{\boldsymbol{T}} t \iff_{\text{def}} \boldsymbol{T}(s) = \boldsymbol{T}(t)\,.$$

Two μ-terms $s, t \in T_\mu(\Sigma)$ are called *equivalent* if and only if $s =_{\boldsymbol{T}} t$.

We can now recognise that the two pairs of μ-terms used in Figure 5 that, respectively, have the same tree unwinding are equivalent, and write this as $s_1 =_{\boldsymbol{T}} t_1$ and $s_2 =_{\boldsymbol{T}} t_2$.

It is well known that the tree unwinding of a μ-term is invariant under the *folding* and *unfolding* operations, which can be formalised as the rewrite relations $\rightarrow_{\text{unfold}}$ and $\rightarrow_{\text{fold}}$

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

defined by the rewrite rules

$$\mu\alpha.\, s \;\to_{\text{unfold}}\; s[\mu\alpha.\, s/\alpha]$$
$$t[\mu\beta.\, t/\beta] \;\to_{\text{fold}}\; \mu\beta.\, t$$

(for all $\alpha, \beta \in \mathit{Var}$ and $s, t \in T_\mu(\Sigma)$). Using the rewriting relation $\to_{\text{unfold}}$ and the concept of an 'infinite normal form' from infinitary rewriting, the tree unwinding of a μ-term s can also be viewed as the infinite normal form with respect to $\to_{\text{unfold}}$ of s, that is, as the possibly infinite term t with the following properties:

— t is the limit of a possibly infinite rewrite sequence consisting only of $\to_{\text{unfold}}$-steps starting from s.
— No $\to_{\text{unfold}}$-step is possible from t (t does not contain an $\to_{\text{unfold}}$-'redex').

Another well-known fact about μ-terms is that, for all $t \in T_\mu(\Sigma)$ (and even for all $t \in \mathit{Ter}_\mu(\Sigma)$), only a finite number of μ-terms can be reached by successive applications of the 'unfolding at the outermost position' and 'decomposition of a μ-term $F(t_1, \ldots, t_n)$ into one of its subterms $t_1, \ldots\ t_n$' operations. In order to formulate this fact about 'generated subterms' of a μ-term, we introduce, in addition to folding and unfolding, a couple of further rewrite rewrite relations that will also turn out to be useful for other purposes.

The rewrite relations:

$\to_{\text{ou}}$ for *outermost-unfolding*,
$\to_{\text{unfold}}$ for *unfolding*,
$\to_{\text{fold}}$ for *folding*,
$\to_{\text{od(i)}}$ for *outermost-decomposition selecting the i-th argument*,
$\to_{\text{od}}$ for *outermost-decomposition*,
$\to_{\text{oud}}$ for *outermost-unfolding or -decomposition*

are defined as subsets of $T_\mu(\Sigma) \times T_\mu(\Sigma)$ as follows:

$$\to_{\text{ou}} =_{\text{def}} \{\langle \mu\alpha.\, t,\, t[\mu\alpha.\, t/\alpha]\rangle \mid \alpha \in \mathit{Var},\, t \in T_\mu(\Sigma)\}$$
$$\to_{\text{unfold}} =_{\text{def}} \{\langle C[t],\, C[s]\rangle \mid C \text{ context},\, C[t], t, s \in T_\mu(\Sigma),\, t \to_{\text{ou}} s\}$$
$$\to_{\text{fold}} =_{\text{def}} \leftarrow_{\text{unfold}}$$
$$\to_{\text{od(i)}} =_{\text{def}} \{\langle F(t_1, \ldots, t_n),\, t_i\rangle \mid n \in \omega,\, F \in \Sigma_n,\, t_1, \ldots, t_n \in T_\mu(\Sigma)\} \quad (\text{for all } i \in \omega)$$
$$\to_{\text{od}} =_{\text{def}} \bigcup_{i \in \omega} \to_{\text{od(i)}}$$
$$\to_{\text{oud}} =_{\text{def}} \to_{\text{ou}} \cup \to_{\text{od}}$$

For any of the rewrite relations $\to_{(\cdot)}$ defined here, we use:

$\leftarrow_{(\cdot)}$ to denote the converse relation;
$\to^{+}_{(\cdot)}$ to denote the transitive closure of $\to_{(\cdot)}$; and
$\twoheadrightarrow_{(\cdot)}$ for the reflexive and transitive closure of $\to_{(\cdot)}$, which is also called the *more-step rewrite relation* with respect to $\to$.

For example, $\twoheadrightarrow_{\text{oud}}$ denotes the more-step rewrite relation with respect to $\to_{\text{oud}}$.

We now define 'generated subterms' of μ-terms. For all $s, t \in T_\mu(\Sigma)$, we say that s is a *generated subterm* of t, denoted $s \sqsubseteq t$, if and only if $t \twoheadrightarrow_{\text{oud}} s$. We use $GS(t)$ to denote the

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

set of all generated subterms of a μ-term t in $T_\mu(\Sigma)$. The following lemma formulates the well-known fact concerning μ-terms that we mentioned above (for example, see Brandt and Henglein (1998), where the term 'syntactical subterm' is used).

Lemma 2.6. For all $t \in T_\mu(\Sigma)$, the set $GS(t)$ of generated subterms of t is finite.

We now state a technical property of generated subterms that will be used later. Suppose that $t \sqsubseteq s \sqsubseteq t$ and $t \to_{\mathrm{ou}} s$ (t is a generated subterm of itself, which is witnessed by a $\to_{\mathrm{oud}}$-rewrite sequence $t \to_{\mathrm{ou}} s \twoheadrightarrow_{\mathrm{oud}} t$ containing an $\to_{\mathrm{ou}}$-step at the start). Then every rewrite sequence that witnesses $s \twoheadrightarrow_{\mathrm{oud}} t$ contains at least one $\to_{\mathrm{od}}$-step. This is an immediate consequence of the following lemma, which states that $\to_{\mathrm{ou}}$ *is strongly normalising*: every rewriting sequence consisting of contiguous $\to_{\mathrm{ou}}$-steps is finite.

Lemma 2.7. The rewrite relation $\to_{\mathrm{ou}}$ is strongly normalising. Consequently there does not exist a μ-term $t \in T_\mu(\Sigma)$ such that $t \to^+_{\mathrm{ou}} t$.

Proof. Outermost unfolding steps on μ-terms in $T_\mu(\Sigma)$ strictly decrease the number $\mathrm{nl}\mu\mathrm{b}$ of leading μ-bindings: if $s_1 \to_{\mathrm{ou}} s_2$ for $s_1, s_2 \in T_\mu(\Sigma)$, then $\mathrm{nl}\mu\mathrm{b}(s_1) > \mathrm{nl}\mu\mathrm{b}(s_2)$. □

Note that the extension of $\to_{\mathrm{ou}}$ to a rewrite relation on $Ter_\mu(\Sigma)$ is actually not strongly normalising: a cyclic rewrite sequence such as $\mu\alpha\beta.\,\alpha \to_{\mathrm{ou}} \mu\beta\alpha\beta.\,\alpha \to_{\mathrm{ou}} \mu\alpha\beta.\,\alpha$ becomes possible (in the first step the number of leading μ-bindings is actually increased).

The next lemma states that, for every subtree $\boldsymbol{T}(t)|_p$ of the tree unwinding $\boldsymbol{T}(t)$ on a μ-term t, there exists a generated subterm t_p of t that has $\boldsymbol{T}(t)|_p$ as its tree unwinding. This technical statement will be needed later in the proof of Theorem 2.13.

Lemma 2.8. $(\forall t \in T_\mu(\Sigma))\,(\forall p \in Acc(t))\,(\exists\, t_p \in GS(t))\,\big[\,\boldsymbol{T}(t_p) = \boldsymbol{T}(t)|_p\,\big]$.

Sketch of proof. The lemma is a consequence of the fact that, in the definition of the tree unwinding, the stipulation for $\boldsymbol{T}(t)(p)$, for $t \in Ter_\mu(\Sigma)$ and $p \in \omega^*$, only recurs on $\boldsymbol{T}(t')(p')$ for generated subterms t' of t, and for paths p' that are shorter than or equal in length to that of p. This fact can be used to show the lemma by induction on the length $|p|$ of p with a subinduction on the number of leading μ-bindings in t. □

We proceed by defining a concept of bisimilarity between μ-terms that is closely related to tree unwinding equivalence. In fact, we will show later that if two μ-terms are bisimilar according to the definition below, then they are equivalent. For a binary relation R on $T_\mu(\Sigma)$ to be a bisimulation, three conditions are required to hold for all pairs $\langle s, t\rangle \in R$:

1 s and t must have the same leading symbol.
2 The results of $\twoheadrightarrow_{\mathrm{ou}}$-steps from s and t must also be related by R.
3 For all $i \in \omega$, if s and t rewrite by $\to_{\mathrm{od}(i)}$-steps, then the reducts must again be related via R.

Definition 2.9. A non-empty relation $R \subseteq T_\mu(\Sigma) \times T_\mu(\Sigma)$ is a *bisimulation on μ-terms* (*on* $T_\mu(\Sigma)$) if and only if the following three conditions are fulfilled for all $\langle s, t\rangle \in R$:

(i) $\mathscr{L}(s) = \mathscr{L}(t)$.
(ii) If $s \twoheadrightarrow_{\mathrm{ou}} s'$ and $t \twoheadrightarrow_{\mathrm{ou}} t'$, for $s', t' \in T_\mu(\Sigma)$, then $\langle s', t'\rangle \in R$.

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

(iii) If $s \to_{\mathrm{od(i)}} s'$ and $t \to_{\mathrm{od(i)}} t'$, for some $i \in \omega$ and $s', t' \in T_\mu(\Sigma)$, then $\langle s', t' \rangle \in R$ (that is, if $s \equiv F(s_1, \ldots, s_n)$ and $t \equiv G(t_1, \ldots, t_m)$, then $\langle s_i, t_i \rangle \in R$ must hold for all $i \in \{1, \ldots, \min\{n, m\}\}$).

For $s, t \in T_\mu(\Sigma)$ we write $s \sim t$ if there exists a bisimulation R on $T_\mu(\Sigma)$ with $\langle s, t \rangle \in R$; if there exists a finite bisimulation R with $\langle s, t \rangle \in R$, we write $s \sim_{\mathrm{fin}} t$.

The proposition below states an easy property of bisimulations on μ-terms. Item (ii) can be proved using Lemma 2.6. The subsequent lemma is an immediate consequence.

Proposition 2.10. Let $s, t \in T_\mu(\Sigma)$ and R be a bisimulation on $T_\mu(\Sigma)$ with $\langle s, t \rangle \in R$. Then:

(i) For all generated subterms s' of s there exists a generated subterm t' of t such that $\langle s', t' \rangle \in R$, and *vice versa*.
(ii) The relation $R \cap (GS(s) \times GS(t))$ is a finite bisimulation on $T_\mu(\Sigma)$ with $\langle s, t \rangle \in R$.

Lemma 2.11. For all $s, t \in T_\mu(\Sigma)$, we have $s \sim t \iff s \sim_{\mathrm{fin}} t$.

The next lemma states that if two μ-terms are bisimilar, they are also equivalent.

Lemma 2.12. If R is a bisimulation on $T_\mu(\Sigma)$, then $s =_{\boldsymbol{T}} t$ for all $\langle s, t \rangle \in R$.

Sketch of proof. Suppose that R is a bisimulation on $Ter_\mu(\Sigma)$. The statement

$$(\forall \langle s, t \rangle \in R)\,(\forall p \in \omega^*)\,[\,\boldsymbol{T}(s)(p) = \boldsymbol{T}(t)(p)\,]$$

can be shown by an induction on the length $|p|$ of p with a subinduction on $\mathrm{nl}\mu\mathrm{b}(s) + \mathrm{nl}\mu\mathrm{b}(t)$, the sum of the number of leading μ-bindings of s and t. □

In the proof of the next theorem, and later, we will encounter binary relations R on $T_\mu(\Sigma)$ that are almost bisimulations in the sense that only a (finite) number of pairs are missing, each of which can be reached from a pair $\langle s, t \rangle \in R$ by performing multiple $\to_{\mathrm{ou}}$-steps to s and to t. We will introduce a specific name for such relations: a relation $R \subseteq T_\mu(\Sigma) \times T_\mu(\Sigma)$ is said to be a *bisimulation on $T_\mu(\Sigma)$ up to adding $\to_{ou}$-reachable pairs* if and only if the extension

$$\tilde{R} =_{\mathrm{def}} \{\, \langle s', t' \rangle \mid (\exists \langle s, t \rangle \in R)\,[\, s \twoheadrightarrow_{\mathrm{ou}} s' \;\&\; t \twoheadrightarrow_{\mathrm{ou}} t' \,]\,\} \tag{2.1}$$

of R is a bisimulation on $T_\mu(\Sigma)$.

The following theorem now establishes that the concept of bisimilarity between μ-terms introduced in Definition 2.9 coincides with tree unwinding equivalence. Furthermore, it formulates a 'finite bisimulation principle' for proving the equivalence of μ-terms.

Theorem 2.13. For all $s, t \in T_\mu(\Sigma)$:

$$s \sim t \implies s =_{\boldsymbol{T}} t \tag{2.2}$$

$$s =_{\boldsymbol{T}} t \implies s \sim_{\mathrm{fin}} t\,. \tag{2.3}$$

Thus, to prove that two μ-terms $s, t \in T_\mu(\Sigma)$ are equivalent, it suffices to find a bisimulation on $T_\mu(\Sigma)$ that contains $\langle s, t \rangle$. If $s, t \in T_\mu(\Sigma)$ are equivalent, then even a finite bisimulation

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

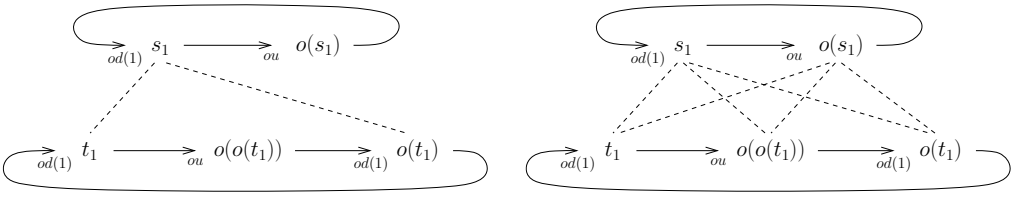

Figure 6. A bisimulation R on $T_\mu(\Sigma)$ up to adding $\rightarrow_{\mathrm{ou}}$-reachable pairs (left) and its extension $\tilde{R}$ to a bisimulation on $T_\mu(\Sigma)$ (right) relating the μ-terms $s_1 \equiv \mu\alpha.\, o(\alpha)$ and $t_1 \equiv \mu\beta.\, o(o(\beta))$.

containing $\langle s, t\rangle$ can be found. As a consequence, all three of the binary relations $\sim$, $\sim_{\mathrm{fin}}$ and $=_{\boldsymbol{T}}$ on $T_\mu(\Sigma)$ coincide.

Proof. It suffices to show (2.2) and (2.3) for all $s, t \in T_\mu(\Sigma)$. That (2.2) holds for all $s, t \in T_\mu(\Sigma)$ follows immediately from Lemma 2.12.

To prove the second statement, let $s, t \in T_\mu(\Sigma)$ be arbitrary with $s =_{\boldsymbol{T}} t$. Because of Lemma 2.8, we can choose, for all $p \in \omega^*$, generated subterms $s_p \in GS(s)$ and $t_p \in GS(t)$ such that $\boldsymbol{T}(s_p) = \boldsymbol{T}(s)|_p$ and $\boldsymbol{T}(t_p) = \boldsymbol{T}(t)|_p$. With these μ-terms, we let $R = \{\langle s_p, t_p\rangle | \; p \in \omega^*\}$. Obviously, we have $\langle s, t\rangle \in R$ and $R \subseteq GS(s) \times GS(t)$. It is not difficult to verify that R is a bisimulation up to adding $\rightarrow_{\mathrm{ou}}$-reachable pairs, which means that the extension $\tilde{R}$ of R defined by (2.1) is a bisimulation. Since, by the definition of $\tilde{R}$, we also have $\tilde{R} \subseteq GS(s) \times GS(t)$, it follows from Lemma 2.6 that $\tilde{R}$ is finite. As $\langle s, t\rangle \in \tilde{R}$, we have found a finite bisimulation $\tilde{R}$ with $\langle s, t\rangle \in \tilde{R}$. Therefore we have shown $s \sim_{\mathrm{fin}} t$. □

Example 2.14. Consider the μ-terms $s_1 \equiv \mu\alpha.\, o(\alpha)$ and $t_1 \equiv \mu\beta.\, o(o(\beta))$, which represent the term graphs $\mathscr{G}_1$ and $\mathscr{H}_1$, respectively, in Figure 1. From the cyclic form of the reduction graphs with respect to $\rightarrow_{\mathrm{oud}}$ of s_1 and t_1 (see Figure 6), it is easy to check that $\tilde{R} = \{s_1, o(s_1)\} \times \{t_1, o(t_1), o(o(t_1))\}$ is a finite bisimulation on $T_\mu(\Sigma)$ that relates s_1 and t_1 (see the right-hand picture in Figure 6). Hence $s_1 \sim_{\mathrm{fin}} t_1$. By the finite bisimulation principle, $s_1 =_{\boldsymbol{T}} t_1$ follows, confirming our earlier observation that $\boldsymbol{T}(s_1) = \boldsymbol{T}(t_1)$.

An example of a bisimulation up to adding $\rightarrow_{\mathrm{ou}}$-reachable pairs linking s_1 and t_1 is the relation $R = \{\langle s_1, t_1\rangle, \langle s_1, o(t_1)\rangle\}$ (illustrated by the left-hand picture in Figure 6).

2.2. *Preliminaries for canonical term graph specifications*

In this subsection we define canonical term graph specifications (ctgs's) and the concept of bisimilarity between ctgs's. Reasoning about bisimilarity of ctgs's will be formalised later in the proof systems defined in Section 6.

In order to keep the technicalities to a minimum, we follow Ariola and Klop and consider only equational specifications of cyclic term graphs without free variables (Ariola and Klop 1995). (All of the results in this section can be generalised straightforwardly to the case of equational specifications with free variables.)

We assume that a countably infinite set *RVar* of *recursion variables* underlies the following definition. Just as for variables in μ-terms, we use small Greek letters $\alpha, \beta, \ldots$ for recursion variables. We again use $\equiv$ to denote syntactical equality between terms.

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

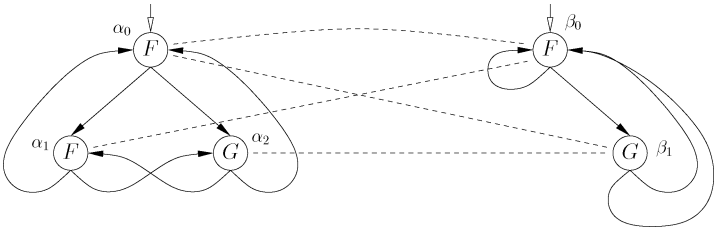

Figure 7. The bisimilar term graphs $\mathscr{G}_3$ and $\mathscr{H}_3$, where F and G are binary function symbols.

Definition 2.15. Let Σ be a signature. A *canonical term graph specification* (a *ctgs*) is an equational specification of the form $\langle\, \alpha_0 \,|\, \{\alpha_0 = t_0, \ldots, \alpha_n = t_n\} \,\rangle$, where $n \in \omega$, $\alpha_0, \ldots, \alpha_n$ are distinct recursion variables in $RVar$ and, for all $i \in \{0, 1, \ldots, n\}$, 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 recursion variables $\alpha_{i1}, \ldots, \alpha_{in_i} \in \{\alpha_0, \ldots, \alpha_n\}$; the (recursion) equation $\alpha_0 = t_0$ is called the *leading equation* of the specification, α_0 the *leading variable* and t_0 the *leading term*. We will use the letters g and h to vary through ctgs's, and the letters $\mathscr{E}$ and $\mathscr{F}$ for sets of recursion equations. For a ctgs g, we use $lv(g)$ to denote the leading variable of g. Finally, we use $\mathscr{TGS}(\Sigma)$ to denote the set of all ctgs's over Σ.

It is straightforward to define a notion of tree unwinding for ctgs's: every finite path that is possible in a ctgs $g = \langle\, \alpha_0 \,|\, \mathscr{E}_g \,\rangle$ from the leading variable α_0, via transitions specified by $\mathscr{E}_g$, to a recursion variable of $\mathscr{E}_g$ corresponds uniquely to an 'access path' in the tree unwinding of g. We will not introduce this notion formally here, but mention that it provides a reason for the significance of the concept of 'bisimilarity' for ctgs's defined below. This is because ctgs's g and h have the same tree unwinding if and only if g and h are 'bisimilar' (Ariola and Klop 1995).

Definition 2.16. Let Σ be a signature and $g = \langle\, \alpha_0 \,|\, \{\alpha_0 = t_0, \ldots, \alpha_n = t_n\} \,\rangle$ and $h = \langle\, \alpha'_0 \,|\, \{\alpha'_0 = t'_0, \ldots, \alpha'_{n'} = t'_{n'}\} \,\rangle$ be canonical term graph specifications over Σ.

A binary relation $R \subseteq \{\alpha_0, \ldots, \alpha_n\} \times \{\alpha'_0, \ldots, \alpha'_{n'}\}$ is called a *bisimulation* between g and h if and only if the following two conditions hold:

(i) $\langle \alpha_0, \alpha'_0 \rangle \in R$.

(ii) If $\langle \alpha_i, \alpha'_j \rangle \in R$ for some i, j with $0 \leqslant i \leqslant n$ and $0 \leqslant j \leqslant n'$, and if (with some $n_i, n'_j \in \omega$), we have $t_i \equiv F(\alpha_{i1}, \ldots, \alpha_{in_i})$ and $t'_j \equiv F'(\alpha'_{j1}, \ldots, \alpha'_{jn'_j})$, then $F \equiv F'$ (and hence $n_i = n'_j$) and $\langle \alpha_{i1}, \alpha'_{j1} \rangle \in R, \ldots, \langle \alpha_{in_i}, \alpha'_{jn'_j} \rangle \in R$ must also hold.

We say that g and h are *bisimilar* (denoted symbolically by $g \leftrightarroweq h$) if and only if there exists a bisimulation between g and h.

Example 2.17. Consider the canonical term graph specifications

$$g = \langle\, \alpha_0 \,|\, \mathscr{E}_g \,\rangle = \langle\, \alpha_0 \,|\, \{\alpha_0 = F(\alpha_1, \alpha_2),\ \alpha_1 = F(\alpha_0, \alpha_2),\ \alpha_2 = G(\alpha_1, \alpha_0)\} \,\rangle \tag{2.4}$$

$$h = \langle\, \beta_0 \,|\, \mathscr{E}_h \,\rangle = \langle\, \beta_0 \,|\, \{\beta_0 = F(\beta_0, \beta_1),\ \beta_1 = G(\beta_0, \beta_0)\} \,\rangle \tag{2.5}$$

in $\mathscr{TGS}(\{F, G\})$, where F and G are binary function symbols. These ctgs's correspond to the cyclic term graphs $\mathscr{G}_3$ and $\mathscr{H}_3$, respectively, in Figure 7. It is easy to verify that

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

the relation $R = \{\langle \alpha_0, \beta_0\rangle, \langle \alpha_1, \beta_0\rangle, \langle \alpha_2, \beta_1\rangle\}$ is a bisimulation between g and h according to Definition 2.16 (for the cyclic term graphs $\mathscr{G}_3$ and $\mathscr{H}_3$ represented in Figure 7, this relation is shown by the broken lines, which connect nodes corresponding to the recursion variables of g and h). This shows that $g \leftrightarroweq h$.

3. Brandt and Henglein's coinductive proof system BH

In this section we define adaptations **BH**(Σ) of Brandt and Henglein's coinductive axiomatisation of recursive type equivalence (Brandt and Henglein 1998) for tree unwinding equivalence on μ-terms over signature Σ. We give an example of a derivation in such a system, and formulate a completeness theorem for **BH**(Σ) systems.

The **BH**(Σ) systems we define here are straightforward generalisations of Brandt and Henglein's system, which is the special case of a **BH**(Σ) system with the signature $\Sigma = \{\bot, \top, \rightarrow\}$ (where $\bot$ and $\top$ are constants that denote the bottom and top types, respectively, and $\rightarrow$ is the construction symbol for composition types). Unlike Brandt and Henglein's formalisation of their system as a Gentzen-style sequent calculus, we define **BH**(Σ) systems here as natural-deduction style systems based on the format of 'N-systems', as described in Troelstra and Schwichtenberg (2000).

A characteristic feature of derivations in natural-deduction systems is that one is allowed to use assumptions that may be 'closed' (or 'discharged') at a later stage in the derivation, or in the extension of a derivation to a longer derivation. In defining **BH**(Σ) systems, we assume a countably infinite set *Mk* of assumption markers, which are used as bookmarking devices for the bindings of assumptions to rule applications at which they are discharged. Here, and later, we will use u, v and w as syntactical variables for assumption markers.

Definition 3.1. The natural-deduction style proof system **BH**(Σ) allows equations $s = t$ with $s, t \in T_\mu(\Sigma)$ as its *formulas*. It contains the *axioms* (that is, *zero-premise rules*) REFL and FOLD/UNFOLD, allows *assumptions* (Assm), and contains the *inference rules* TRANS, SYMM and c-COMP listed in Figure 8. The rule c-COMP enables applications at which assumptions of the form of the conclusion are discharged (c-COMP is the only rule of **BH**(Σ) with assumption discharging applications). To simplify our notation, from now on we will keep the underlying signature implicit and write **BH** instead of **BH**(Σ).

As a motivation for the definition of derivations in **BH** given below, consider the example shown in Figure 9 of the derivation $\mathscr{D}$ in **BH** with the equation $\mu\alpha.\, o(\alpha) = \mu\beta.\, o(o(\beta))$ as its conclusion. Recall that $\mu\alpha.\, o(\alpha)$ and $\mu\beta.\, o(o(\beta))$ are representations (via Ariola and Klop's translation $\mathscr{G}$) of the bisimilar cyclic term graphs $\mathscr{G}_1$ and $\mathscr{H}_1$ in Figure 1. The derivation $\mathscr{D}$ contains a single assumption at its top: the marked formula $(o(s) = o(o(t)))^u$. It is this formula that is discharged in $\mathscr{D}$ at the bottommost application of the 'circular decomposition rule' c-COMP; this fact is indicated by attaching the assumption marker u of the discharged assumption to the label of the application of c-COMP at which it is discharged. The derivation $\mathscr{D}$ is a close counterpart of the Brandt–Henglein derivation shown on the right-hand side of Figure 2, where equations between μ-terms here take over the role of equations between 'leading terms' in term

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

The *axioms* (*zero-premise rules*) and possible *assumptions* of $\mathbf{BH}(\Sigma)$:

$$\text{(REFL)}\quad t = t \qquad\qquad \text{(FOLD/UNFOLD)}\quad \mu\alpha.\, t = t[\mu\alpha.\, t/\alpha]$$

$$\text{(Assm)}\quad (s = t)^u \quad (\text{with } u \in Mk)$$

The *inference rules* of $\mathbf{BH}(\Sigma)$:

$$\frac{\begin{array}{c}\mathcal{D}_1\\ s = t\end{array}}{t = s}\ \text{SYMM} \qquad\qquad \frac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2\\ s = r & r = t\end{array}}{s = t}\ \text{TRANS}$$

$$\frac{\begin{array}{ccc}[F(s_1,\ldots,s_n) = F(t_1,\ldots,t_n)]^u & & [F(s_1,\ldots,s_n) = F(t_1,\ldots,t_n)]^u\\ \mathcal{D}_1 & & \mathcal{D}_n\\ s_1 = t_1 & \cdots & s_n = t_n\end{array}}{F(s_1,\ldots,s_n) = F(t_1,\ldots,t_n)}\ \text{c-COMP},\ u \quad (\text{for } F \in \Sigma_n)$$

Figure 8. The Brandt–Henglein system $\mathbf{BH}(\Sigma)$ for tree unwinding equivalence $=_T$ on $T_\mu(\Sigma)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{s = o(s)}\,(\text{FOLD/UNFOLD})}{s = o(o(t))}\,\text{TRANS}\qquad \dfrac{}{(o(s) = o(o(t)))^u}\,(\text{Assm})}{s = o(t)}\,\text{c-COMP}}{s = o(s)}\,\text{TRANS}\qquad \dfrac{\dfrac{}{t = o(o(t))}\,(\text{FOLD/UNFOLD})}{o(o(t)) = t}\,\text{SYMM}}{s = t}\,\text{TRANS}}{o(s) = o(o(t))}\,\text{c-COMP},\ u}{s = o(o(t))}\,\text{TRANS}\qquad \dfrac{\dfrac{}{t = o(o(t))}\,(\text{FOLD/UNFOLD})}{o(o(t)) = t}\,\text{SYMM}}{\mu\alpha.\, o(\alpha) = \mu\beta.\, o(o(\beta))}\,\text{TRANS}$$

Figure 9. The derivation $\mathscr{D}$ in **BH** without open assumptions of the equation $s = t$, where $s \equiv \mu\alpha.\, o(\alpha)$ and $t \equiv \mu\beta.\, o(o(\beta))$ are the μ-terms representing $\mathscr{G}_1$ and $\mathscr{H}_1$ in Figure 1.

graph specifications in the earlier example. Compared with the easy linear structure of the derivation sketched on the right-hand side of Figure 2, the derivation $\mathscr{D}$ has a more complicated form because performing the outermost-folding operation on a term on either side of an equation in the system **BH** requires an axiom FOLD/UNFOLD, an application of TRANS, and, for outermost-unfolding steps on the right, an application of SYMM.

General definition of derivations in **BH**

A *derivation in* **BH** is a prooftree, that is, a finite upwards-growing tree with nodes that are labelled by formulas, or marked formulas, such that:

— the leaves at the top are labelled by axioms, or by assumptions, which are formulas with a superscript-marker attached to it;
— in assumptions, different markers are attached to different formulas (to distinguish different assumptions by their markers);
— assumptions occurring in a derivation may be 'undischarged' (also called 'open') or 'discharged' (also called 'closed'), see below;

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

— formulas at an internal node v of the prooftree arise through applications of **BH**-rules from the formulas at the immediate successors of v, where (depending on the kind of rule applied) assumptions may be discharged;
— the bottommost formula is called the *conclusion*.

If an assumption A^u is *discharged* (or *closed*) by a rule application in a derivation, this is indicated by writing the marker u next to the rule name label at this application (markers allow us to identify which assumptions get 'discharged' at a rule application).

An occurrence of an assumption A^u in a derivation $\mathscr{D}$ is said to be *undischarged* (or *open*) in $\mathscr{D}$ if and only if on the path down to the conclusion of $\mathscr{D}$ no assumption is passed at which A^u is discharged; otherwise the occurrence of this assumption is called *discharged* (or *closed*) in $\mathscr{D}$. Occurrences of open assumptions in a derivation $\mathscr{D}$ that are occurrences of the same marker attached to the same formula form together an *open assumption set* of $\mathscr{D}$.

As usual for natural-deduction systems, theorems of the system **BH** are now defined as conclusions of derivations without open assumptions: a formula $s = t$ is a *theorem* of **BH**, denoted $\vdash_{\mathbf{BH}} s = t$, if and only if there is a derivation in **BH** without open assumptions and with conclusion $s = t$.

The following soundness and completeness theorem holds for **BH** with respect to tree unfolding equivalence.

Theorem 3.2. The proof system **BH** is sound and complete with respect to tree unwinding equivalence, that is, for all $s, t \in T_\mu(\Sigma)$,

$$\vdash_{\mathbf{BH}} s = t \quad \Longleftrightarrow \quad s =_{\boldsymbol{T}} t \ . \tag{3.6}$$

Proof hint. The proof is an adaptation of the argument given by Brandt and Henglein for the soundness and completeness of their coinductive axiomatisation for the subtyping relation on recursive types (Brandt and Henglein 1998).

Alternatively, for the completeness direction, '$\Leftarrow$' in (3.6), we can apply the finite bisimulation principle, Theorem 2.13, to obtain, for $s, t \in T_\mu(\Sigma)$ with $s =_{\boldsymbol{T}} t$, a finite bisimulation R with $\langle s, t\rangle \in R$. It is then straightforward to extract a derivation in **BH** with conclusion $s = t$ and no open assumptions from such a finite bisimulation. □

4. Ariola and Klop's proof system AK for bisimilarity checking

In this section we introduce an adaptation **AK**(Σ) of Ariola and Klop's 'syntactic-matching' proof system (Ariola and Klop 1995) for tree unwinding equivalence on μ-terms over a general signature Σ. We define 'contradictions with respect to $=_{\boldsymbol{T}}$' and the concept of 'consistency with respect to **AK**', and formulate a correspondence theorem between consistency with respect to **AK** and tree unwinding equivalence, which is the basis for the usefulness of this system. The most conspicuous feature of this system is the decomposition rule DECOMP, which is a 'destructive' counterpart of the 'constructive' composition rule c-COMP of the **BH** system.

Definition 4.1. The Hilbert-style proof system **AK**(Σ) contains the equations between μ-terms in $T_\mu(\Sigma)$ over Σ as its *formulas*. Its *axioms* and *inference rules* are given in

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

The *axioms* of $\mathbf{AK}(\Sigma)$:

$$\text{(REFL)}\quad t = t \qquad\qquad \text{(FOLD/UNFOLD)}\quad \mu\alpha.\, t = t[\mu\alpha.\, t/\alpha]$$

The *inference rules* of $\mathbf{AK}(\Sigma)$:

$$\frac{\mathcal{D}_1}{\dfrac{s = t}{t = s}}\ \text{SYMM} \qquad\qquad \frac{\dfrac{\mathcal{D}_1}{s = r}\quad \dfrac{\mathcal{D}_2}{r = t}}{s = t}\ \text{TRANS}$$

$$\frac{F(s_1, \ldots, s_n) = F(t_1, \ldots, t_n)}{s_i = t_i}\ \text{DECOMP} \quad (\text{for } F \in \Sigma_n \text{ and } i \in \{1, \ldots, n\})$$

Figure 10. The Ariola–Klop system $\mathbf{AK}(\Sigma)$ for tree unwinding equivalence on $T_\mu(\Sigma)$.

Figure 10: the formulas belonging to the schemes REFL and FOLD/UNFOLD are the axioms, and SYMM, TRANS and DECOMP are the rules of $\mathbf{AK}(\Sigma)$. As with $\mathbf{BH}(\Sigma)$, we generally abbreviate $\mathbf{AK}(\Sigma)$ to **AK**.

For all $s, t \in T_\mu(\Sigma)$, we use $s = t \vdash_{\mathbf{AK}} s_1 = t_1$ to denote the assertion that there is a derivation in **AK** from the assumption $s = t$ that has conclusion $s_1 = t_1$.

In order to formulate a statement linking a concept of 'relative consistency with respect to **AK**' to tree unwinding equivalence and bisimilarity of μ-terms, we need to stipulate when a formula of **AK** is to be called 'consistent' with respect to this system. To do this, we first need to define what we mean by 'contradictions with respect to $=_{\boldsymbol{T}}$'.

Definition 4.2. An equation $s = t$, where $s, t \in T_\mu(\Sigma)$, is called a *contradiction with respect to* $=_{\boldsymbol{T}}$ if and only s and t have different leading symbols.

Thus, a contradiction $s = t$ with respect to $=_{\boldsymbol{T}}$ is an equation between the μ-terms $s, t \in T_\mu(\Sigma)$ for which it is obvious that s and t are not equivalent: their respective tree unwindings already differ in the symbols labelling the roots. Examples of contradictions are the equations $\alpha = \beta$ (if $\alpha \not\equiv \beta$), $C = \alpha$, and $\mu\alpha.\, F(\alpha, \beta) = \mu\gamma.\, \delta$. On the other hand, for all $s, t, s_1, t_1 \in T_\mu(\Sigma)$, $F(s, t) = \mu\alpha.\, F(s_1, t_1)$ is not a contradiction with respect to $=_{\boldsymbol{T}}$.

Definition 4.3. Let $s, t \in T_\mu(\Sigma)$. The equation $s = t$ is called **AK**-*inconsistent* if and only if we have $s = t \vdash_{\mathbf{AK}} s_1 = t_1$ for a contradiction $s_1 = t_1$ with respect to $=_{\boldsymbol{T}}$; otherwise it is called **AK**-*consistent* or *consistent with* **AK**.

We can now formulate the correspondence theorem between consistency with respect to **AK** and tree unwinding equivalence.

Theorem 4.4. Consistency with respect to **AK** corresponds to tree unwinding equivalence in the following sense: for all $s, t \in T_\mu(\Sigma)$,

$$s = t \ \text{ is } \mathbf{AK}\text{-consistent} \iff s =_{\boldsymbol{T}} t \ . \tag{4.7}$$

Proof hint. The proof is an adaptation of the proof given by Ariola and Klop for their syntactic-matching system (Ariola and Klop 1995), taking advantage of the feature of the decomposition rule that enables us to carry out repeated 'experiments' that take simultaneous 'looks' into component subterms on either side of an equation such as

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

$F(s_1,\ldots,s_n) = F(t_1,\ldots,t_n)$, and other equations, to yield an 'observation' such as $s_i = t_i$, for some $i \in \{1,\ldots,n\}$. Using this and the relationship to the tree unwinding, it can be shown that a contradiction with respect to $=_{\boldsymbol{T}}$ is derivable in **AK** from an equation $s = t$, for $s, t \in T_\mu(\Sigma)$, if and only if there exists an access path p of s or of t such that $\boldsymbol{T}(s)(p) \neq \boldsymbol{T}(t)(p)$. □

5. The $\mathbf{BH_0}$ and $\mathbf{AK_0}$ variant systems

In this section we define the variants of the **BH** and **AK** proof systems on which our results will be based. These variant systems, which will be called $\mathbf{BH_0}$ and $\mathbf{AK_0}$, have stronger proof-theoretic properties than **BH** and **AK**, but allow analogous completeness or correspondence theorems. Finally, we give a conservative extension **e-BH$_0$** of $\mathbf{BH_0}$ that has two further inference rules that formalise 'circular' coinductive reasoning.

The presence of symmetry and transitivity rules in the **BH** and **AK** systems provides great flexibility for finding derivations in these systems. But, as a consequence, these rules contribute to a major proof-theoretic disadvantage: for all $s, t \in T_\mu(\Sigma)$, there is a very complex search space for a derivation $\mathscr{D}$ in **BH** with conclusion $s = t$ and no open assumptions, or dually, for surveying all possible derivations in **AK** from the assumption $s = t$ to check whether these contain contradictions. Furthermore, it is not clear how consistency checks in **AK** might be captured conceptually, since the same assumption may be used several times in a derivation in **AK**. This was not the case for the consistency checks discussed in Section 1, where it was assumed that the underlying syntactic-matching system does not contain symmetry and transitivity rules.

It is for these reasons that we now introduce variants of **BH** and **AK** that enjoy the same completeness properties as the original systems, but do not contain symmetry and transitivity rules. We start by defining the $\mathbf{BH_0}$ variant of **BH**.

Definition 5.1. The natural-deduction style proof system $\mathbf{BH_0}(\Sigma)$ allows equations $s = t$ between μ-terms in $T_\mu(\Sigma)$ as its *formulas*. It contains the *axioms* REFL, allows *assumptions* (Assm) and contains the *inference rules* FOLD_l, FOLD_r, COMP and c-COMP listed in Figure 11. The c-COMP rule enables applications at which assumptions are discharged. In fact, some assumptions must be discharged by applications of c-COMP as stipulated by the *side-condition* **I**: with reference to the designations used for the schematic application of this rule displayed in Figure 11, the set of all assumptions $(\, F(s_1,\ldots,s_n) = F(t_1,\ldots,t_n)\,)^u$ that are open in one of $\mathscr{D}_1, \ldots, \mathscr{D}_n$, which are discharged by the application of c-COMP at the bottom, must be non-empty.

Derivations in $\mathbf{BH_0}(\Sigma)$ are defined analogously to derivations in $\mathbf{BH}(\Sigma)$. A formula $s = t$ is a *theorem* of $\mathbf{BH_0}(\Sigma)$, denoted by $\vdash_{\mathbf{BH_0}(\Sigma)} s = t$, if and only if there is a derivation in $\mathbf{BH_0}(\Sigma)$ without open assumptions and with conclusion $s = t$.

As with $\mathbf{BH}(\Sigma)$, we will generally abbreviate $\mathbf{BH_0}(\Sigma)$ to $\mathbf{BH_0}$. We will also use $\mathrm{FOLD}_{l/r}$ to mean both or either of the FOLD_l and FOLD_r rules: for example, we will write '... holds for a $\mathrm{FOLD}_{l/r}$ rule', to mean '... holds for a FOLD_l or FOLD_r rule', and '... holds for $\mathrm{FOLD}_{l/r}$ rules' to mean '... holds for the FOLD_l and FOLD_r rules'.

We have split the coinductive composition rule c-COMP of **BH** into two parts for $\mathbf{BH_0}$: a restricted version of c-COMP with applications at which assumptions have to

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

The *axioms* and possible *assumptions* in $\mathbf{BH_0}(\Sigma)$:

$$\text{(REFL)}\quad t = t \qquad\qquad \text{(Assm)}\quad (s = t)^u \quad (\text{with } u \in Mk)$$

The *inference rules* of $\mathbf{BH_0}(\Sigma)$:

$$\frac{\mathcal{D}_1}{\dfrac{s[\mu\alpha.\, s/\alpha] = t}{\mu\alpha.\, s = t}\ \text{FOLD}_l} \qquad \frac{\mathcal{D}_1}{\dfrac{s = t[\mu\beta.\, t/\beta]}{s = \mu\beta.\, t}\ \text{FOLD}_r} \qquad \frac{\begin{matrix}\mathcal{D}_1 & & \mathcal{D}_n\end{matrix}}{\dfrac{\begin{matrix}s_1 = t_1 & \dots & s_n = t_n\end{matrix}}{F(s_1,\dots,s_n) = F(t_1,\dots,t_n)}\ \begin{matrix}\text{COMP}\\ (\text{if } F \in \Sigma_n)\end{matrix}}$$

$$\frac{\begin{matrix}[F(s_1,\dots,s_n) = F(t_1,\dots,t_n)]^u & & [F(s_1,\dots,s_n) = F(t_1,\dots,t_n)]^u\\ \mathcal{D}_1 & & \mathcal{D}_n\end{matrix}}{\dfrac{\begin{matrix}s_1 = t_1 & \dots & s_n = t_n\end{matrix}}{F(s_1,\dots,s_n) = F(t_1,\dots,t_n)}\ \begin{matrix}\text{c-COMP},\ u\\ (\text{if } F \in \Sigma_n \text{ and}\\ \text{side-cond. } \mathbf{I})\end{matrix}}$$

Figure 11. The normalised variant system $\mathbf{BH_0}(\Sigma)$ without symmetry and transitivity rules of the Brandt–Henglein system $\mathbf{BH}(\Sigma)$.

$$\dfrac{\dfrac{\dfrac{(\dots)^u}{s = t} \qquad \dfrac{\dfrac{\dfrac{(\dots)^v}{F(s,s) = t} \qquad \dfrac{(\dots)^u}{s = t}}{F(F(s,s),s) = F(t,t)}}{s = F(t,t)}\ \text{c-COMP},\ v}{F(s,s) = F(t,F(t,t))} \qquad \dfrac{\dfrac{\dfrac{(\dots)^u}{s = t} \qquad \dfrac{(\dots)^w}{s = F(t,t)}}{F(s,s) = F(t,F(t,t))} \qquad \dfrac{(\dots)^u}{s = t}\ \text{FOLD}_l,\ \text{FOLD}_r}{\dfrac{F(s,s) = t}{F(F(s,s),s) = F(t,t)}\ \text{c-COMP},\ w}}{\dfrac{F(s,s) = t \qquad\qquad s = F(t,t)}{F(F(s,s),s) = F(t,F(t,t))}\ \text{c-COMP},\ u}$$

$$\dfrac{F(F(s,s),s) = F(t,F(t,t))}{\mu\alpha.\, F(F(\alpha,\alpha),\alpha) = \mu\beta.\, F(\beta,F(\beta,\beta))}\ \text{FOLD}_l,\ \text{FOLD}_r$$

$$\begin{aligned}\text{where}\quad (\dots)^u &\quad \text{stands for} \quad (\,F(F(s,s),s) = F(t,F(t,t))\,)^u\,,\\ (\dots)^v &\quad\ \ \text{for} \quad (\,F(s,s) = F(t,F(t,t))\,)^v\,,\ \text{and}\\ (\dots)^w &\quad\ \ \text{for} \quad (\,F(F(s,s),s) = F(t,t)\,)^w\,.\end{aligned}$$

Figure 12. The derivation $\mathscr{D}$ of $s = t$ in $\mathbf{BH_0}(\Sigma)$ without open assumptions for the μ-terms $s \equiv \mu\alpha.\, F(F(\alpha,\alpha),\alpha)$ and $t \equiv \mu\beta.\, F(\beta,F(\beta,\beta))$, which represent $\mathscr{G}_2$ and $\mathscr{H}_2$ in Figure 3.

be discharged; and the plain composition rule COMP with applications at which no assumptions can be discharged. We have done this purely for convenience as it will make it easier to refer to certain case-distinctions in the definitions and proofs in Section 7.

As an example of a derivation in $\mathbf{BH_0}$, consider the derivation $\mathscr{D}$ without open assumptions of the equation $\mu\alpha.\, F(F(\alpha,\alpha),\alpha) = \mu\beta.\, F(\beta,F(\beta,\beta))$ in Figure 12. The μ-terms in this equation are specifications of the cyclic term graphs $\mathscr{G}_2$ and $\mathscr{H}_2$ in Figure 3. Note that the derivation $\mathscr{D}$ is closely related to the looping consistency check in Figure 4: it arises by

(i) mirroring the consistency check at a horizontal line;

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

(ii) replacing the formulas with corresponding μ-terms;
(iii) changing applications of decomposition rules into applications of composition rules COMP;
(iv) adding a few necessary additional applications of $\text{FOLD}_{l/r}$ at the top;
(v) discharging newly arising assumptions at the top at applications of c-COMP below (to achieve this, some applications of decomposition rules COMP have to be changed into applications of c-COMP at this point).

Another example of a $\mathbf{BH_0}$-derivation is shown on the right-hand side of Figure 14.

$\mathbf{BH_0}$ is a 'normalised' version of **BH** in the sense that it has stronger proof-theoretic properties. In particular, $\mathbf{BH_0}$ fulfills the following easily verifiable 'subformula principle', which relates the formulas occurring in a derivation to the conclusion.

Proposition 5.2. Let $\mathscr{D}$ be a derivation in $\mathbf{BH_0}$ with conclusion $s = t$. For all formulas $s_1 = t_1$ in $\mathscr{D}$, s_1 and t_1 are generated subterms of s and t, respectively.

It is easy to see that the absence of symmetry and transitivity rules from $\mathbf{BH_0}$ is crucial for this statement (for transitivity, for example, with distinct $\alpha, \beta, \gamma \in \mathit{Var}$, we can infer $\alpha = \beta$ from $\alpha = \gamma$ and $\gamma = \beta$ by a TRANS-application, where γ is not a generated subterm of either α or β); in particular, **BH** does not enjoy this property. An important consequence of the subformula principle is that proof search in $\mathbf{BH_0}$ is much more restricted, and therefore substantially easier, than in **BH**.

We can prove the following two statements concerning the proof-theoretic relationship between **BH**-derivations and $\mathbf{BH_0}$-derivations in the same way as analogous results can be proved for a pair of corresponding proof systems for recursive type equality (Grabmayer 2005):

— Every derivation $\mathscr{D}$ in $\mathbf{BH_0}$ can be transformed into a derivation $\mathscr{D}'$ in **BH** with the same conclusion and the same open assumptions (this transformation is quite straightforward).
— Conversely, there exists a 'normalisation procedure' that transforms derivations in **BH** into corresponding derivations in $\mathbf{BH_0}$ (this transformation is rather involved).

Derivations in $\mathbf{BH_0}$ are closely related to bisimulations on μ-terms. More precisely, for all derivations $\mathscr{D}$ in $\mathbf{BH_0}$ without open assumptions, the set of all pairs $\langle s, t \rangle$ such that $s = t$ is an equation in $\mathscr{D}$ is a bisimulation up to $\rightarrow_{\text{ou}}$-reachable pairs. This fact is useful when proving the following soundness and completeness theorem for $\mathbf{BH_0}$.

Theorem 5.3. The proof system $\mathbf{BH_0}$ is sound and complete with respect to tree unwinding equivalence, that is, for all $s, t \in T_\mu(\Sigma)$,

$$\vdash_{\mathbf{BH_0}} s = t \quad \Longleftrightarrow \quad s =_{\boldsymbol{T}} t \ . \tag{5.8}$$

Sketch of proof.

'$\Rightarrow$' Suppose that $\mathscr{D}$ is a derivation in $\mathbf{BH_0}$ without open assumptions and with conclusion $s = t$ for some $s, t \in T_\mu(\Sigma)$. If $R = \{\langle \tilde{s}, \tilde{t} \rangle \mid \tilde{s} = \tilde{t} \text{ occurs in } \mathscr{D}\}$, we have $\langle s, t \rangle \in R$. It follows from Proposition 5.2 that $R \subseteq GS(s) \times GS(t)$. It is then straightforward to

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

The *inference rules* of $\mathbf{AK_0}(\Sigma)$:

$$\text{UNFOLD}_l\ \frac{\mu\alpha.\, s = t}{s[\mu\alpha.\, s/\alpha] = t} \qquad \frac{s = \mu\beta.\, t}{s = t[\mu\beta.\, t/\beta]}\ \text{UNFOLD}_r$$

$$\frac{F(s_1, \ldots, s_n) = F(t_1, \ldots, t_n)}{s_i = t_i}\ \text{DECOMP} \quad (\text{for } 1 \leq i \leq n)$$

Figure 13. The normalised variant system $\mathbf{AK_0}(\Sigma)$ without symmetry and transitivity rules of the Ariola–Klop system $\mathbf{AK}(\Sigma)$.

verify that R is a bisimulation up to adding $\to_{\text{ou}}$-reachable pairs, so

$$\tilde{R} = \{\, \langle \tilde{s}_1, \tilde{t}_1 \rangle \mid \langle \tilde{s}, \tilde{t} \rangle \in R,\ \tilde{s} \twoheadrightarrow_{\text{ou}} \tilde{s}_1,\ \tilde{t} \twoheadrightarrow_{\text{ou}} \tilde{t}_1 \} \subseteq GS(s) \times GS(t) \tag{5.9}$$

and $\langle s, t \rangle \in \tilde{R}$ for the extension $\tilde{R}$ of R to a bisimulation. From (5.9), Lemma 2.6 means that $\tilde{R}$ is finite, and the finite bisimulation principle, Theorem 2.13, then implies that $s =_{\boldsymbol{T}} t$.

'$\Leftarrow$' Suppose that $s =_{\boldsymbol{T}} t$, for some $s, t \in T_\mu(\Sigma)$. By the finite bisimulation principle, a finite bisimulation R on $T_\mu(\Sigma)$ exists such that $\langle s, t \rangle \in R$. This bisimulation can be used to show that a straightforward bottom-up proof-search in $\mathbf{BH_0}$ from conclusion $s = t$ yields a derivation in $\mathbf{BH_0}$ with this conclusion and no open assumptions in finitely many steps. □

We continue by defining the $\mathbf{AK_0}$ variant of the syntactic-matching system **AK** defined in the previous section.

Definition 5.4. The Hilbert-style proof system $\mathbf{AK_0}(\Sigma)$ contains the equations between μ-terms in $T_\mu(\Sigma)$ as its *formulas*. It contains *no axioms*. Its *inference rules* are the UNFOLD_l, UNFOLD_r and DECOMP rules listed in Figure 13.

For all $s, t, s_1, t_1 \in T_\mu(\Sigma)$, we use $s = t \vdash_{\mathbf{AK_0}(\Sigma)} s_1 = t_1$ to denote the assertion that there is a derivation in $\mathbf{AK_0}(\Sigma)$ from the assumption $s = t$ that has conclusion $s_1 = t_1$.

We will generally keep the underlying signature Σ implicit by writing $\mathbf{AK_0}(\Sigma)$ for $\mathbf{AK_0}$.

As with $\mathbf{BH_0}$, $\mathbf{AK_0}$ does not contain symmetry or transitivity rules, and it is 'normalised' in a similar sense: $\mathbf{AK_0}$ satisfies the following 'subformula principle', which relates the conclusion of a derivation $\mathscr{D}$ to other formulas of $\mathscr{D}$ in a converse manner to the way it was stated in Proposition 5.2 for derivations in $\mathbf{BH_0}$.

Proposition 5.5. Let $\mathscr{D}$ be a derivation in $\mathbf{AK_0}$ with assumption $s = t$. For all formulas $s_1 = t_1$ in $\mathscr{D}$, s_1 and t_1 are generated subterms of s and t, respectively.

An example of a looping derivation $\mathscr{C}$ in $\mathbf{AK_0}$ is shown on the left-hand side of Figure 14. This derivation $\mathscr{C}$ and the closely related $\mathbf{BH_0}$-derivation $\mathscr{D}$ (on the right-hand side of Figure 14) are formal versions of the derivations in Figure 2 that illustrated our initial observation. The explanations in the Introduction now also apply to the derivations $\mathscr{C}$ and $\mathscr{D}$ here.

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

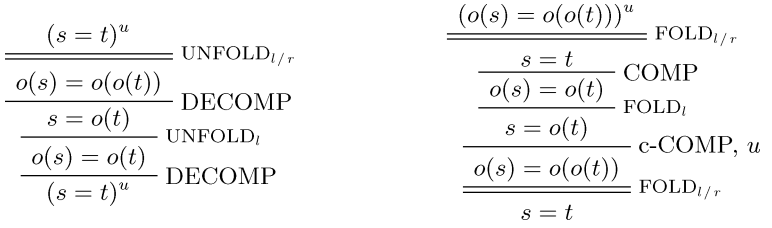

Figure 14. Formalising the proofs in Figure 2 in $\mathbf{AK_0}$ and in $\mathbf{BH_0}$: a derivation $\mathscr{C}$ in $\mathbf{AK_0}$ that is looping (indicated by the repeated formula marker u) and a derivation $\mathscr{D}$ in $\mathbf{BH_0}$, where $s \equiv \mu\alpha.\, o(\alpha)$ and $t \equiv \mu\beta.\, o(o(\beta))$. $\mathscr{D}$ arises from $\mathscr{C}$ by turning it on its head, adding additional folding applications and discharging the new assumption at an application of c-COMP below.

Theorem 5.6. Consistency with respect to $\mathbf{AK_0}$ corresponds to tree unwinding equivalence in the following sense: for all $s, t \in T_\mu(\Sigma)$,

$$s = t \text{ is } \mathbf{AK_0}\text{-consistent} \iff s =_{\boldsymbol{T}} t\,.$$

Proof hint. The proof exploits the fact that a derivation in $\mathbf{AK_0}$ with assumption $s = t$ and conclusion $s_1 = t_1$, for some $s, t, s_1, t_1 \in T_\mu(\Sigma)$, can be viewed as a computation of generated subterms s_1 and t_1 of s and t, respectively, (*cf.* Proposition 5.5) that determines the subtrees $\boldsymbol{T}(s)|p = \boldsymbol{T}(s_1)$ and $\boldsymbol{T}(t)|p = \boldsymbol{T}(t_1)$ of the tree unwindings $\boldsymbol{T}(s)$ and $\boldsymbol{T}(t)$ of s and t at some common access path p. □

There is a slight asymmetry in the relationship via mirroring, as explained in Section 1, between looping consistency checks in an Ariola–Klop system and derivations in a Brandt–Henglein system. This can be seen in the example in Figure 2 as well as from its formalisation for the systems $\mathbf{AK_0}$ and $\mathbf{BH_0}$ in Figure 14: the $\mathbf{BH_0}$-derivation $\mathscr{D}$ on the right arises, after mirroring the looping derivation $\mathscr{C}$ on the left, only by adding new applications of folding (and minor manipulations involving the formula marker u). Conversely, transforming $\mathscr{D}$ into $\mathscr{C}$ requires two rule applications to be discarded before the resulting derivation is then mirrored into a looping derivation corresponding to $\mathscr{C}$. The reason for this asymmetry in the purported relationship between $\mathbf{BH_0}$ and $\mathbf{AK_0}$ is that the 'looping' concept for derivation trees gathering $\mathbf{AK_0}$-derivations is more general than the concept of discharging assumptions in $\mathbf{BH_0}$-derivations. This suggests that an even closer relationship can be established by extending the system $\mathbf{BH_0}$ with rules that also allow the discharge of assumptions in situations where the c-COMP ruleis not applicable. Therefore, we now extend $\mathbf{BH_0}$ by adding two more rules that enable assumption-discharging applications.

Definition 5.7. The extension $\mathbf{e\text{-}BH_0}(\Sigma)$ of the system $\mathbf{BH_0}(\Sigma)$ has the same *formulas* and *axioms* as $\mathbf{BH_0}(\Sigma)$, allows us to make the same *assumptions* and contains all *inference rules* of $\mathbf{BH_0}(\Sigma)$. However, $\mathbf{e\text{-}BH_0}(\Sigma)$ also contains the *inference rules* c-FOLD$_l$ and c-FOLD$_r$ for which schematic applications are shown in Figure 15. Applications of these rules are comparable to applications of the FOLD$_l$ and FOLD$_r$ rules, respectively, but have the additional feature that, as stipulated by the *side-condition* **I**, at least one assumption of

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

$$\dfrac{\begin{array}{c}[\,\mu\alpha.\,s = t\,]^{u}\\ \mathcal{D}_1\\ s[\mu\alpha.\,s/\alpha] = t\end{array}}{\mu\alpha.\,s = t}\ \text{c-FOLD}_l,\ u\ \ (\text{if side-cond. } \mathbf{I}) \qquad \dfrac{\begin{array}{c}[\,s = \mu\beta.\,t\,]^{u}\\ \mathcal{D}_1\\ s = t[\mu\beta.\,t/\beta]\end{array}}{s = \mu\beta.\,t}\ \text{c-FOLD}_r,\ u\ \ (\text{if side-cond. } \mathbf{I})$$

Figure 15. The inference rules c-FOLD$_l$ and c-FOLD$_r$ in the extension **e-BH$_0$**(Σ) of **BH$_0$**(Σ).

the form of the conclusion is discharged. We will again use c-FOLD$_{l/r}$ to mean one or about both of the c-FOLD$_l$ and c-FOLD$_r$ rules, and abbreviate **e-BH$_0$**(Σ) to **e-BH$_0$**.

It is not immediately obvious that the c-FOLD$_{l/r}$ rules formalise sound reasoning with respect to $=_{\mathcal{T}}$. As a step towards showing the soundness of **e-BH$_0$** with respect to $=_{\mathcal{T}}$, we now show that the side-condition **I** on applications of c-FOLD$_{l/r}$ entails the presence of applications of COMP or c-COMP in immediate subderivations.

Lemma 5.8. Let $\mathscr{D}$ be a derivation in **e-BH$_0$** with a bottommost application of the c-FOLD$_l$ or c-FOLD$_r$ rule as shown in Figure 15. Then the subderivation $\mathscr{D}_1$ of $\mathscr{D}$ contains at least one application of the COMP or c-COMP rule.

Proof. Let $\mathscr{D}$ be a derivation in **e-BH$_0$** with a bottommost application of c-FOLD$_l$ as shown in Figure 15. (The proof for a bottommost application of c-FOLD$_r$ is analogous.)

Due to the side-condition **I** for the c-FOLD$_l$-application at the bottom of $\mathscr{D}$, the subderivation $\mathscr{D}_1$ of $\mathscr{D}$ contains at least one open assumption of the form $(\mu\alpha.\,s = t)^u$. We choose an occurrence of such an open assumption at the top of $\mathscr{D}$. By starting at the conclusion and going upwards in $\mathscr{D}$ to the chosen assumption, and thereby looking only at the μ-terms on the left-hand side of the equations encountered, we can construct an $\to_{\text{oud}}$-rewrite sequence from $\mu\alpha.\,s$ back to itself of length greater than or equal to one: passed applications of COMP or c-COMP give rise to single $\to_{\text{od}}$-steps; passed applications of FOLD$_l$ or c-FOLD$_l$ generate $\to_{\text{ou}}$-steps; but passed applications of FOLD$_r$ and c-FOLD$_r$ are ignored. As there is an application of c-FOLD$_l$ at the bottom of $\mathscr{D}$, the resulting $\to_{\text{oud}}$-rewrite sequence from $\mu\alpha.\,s$ back to itself starts with an $\to_{\text{ou}}$-step and therefore has length greater than or equal to one. By Lemma 2.7 it follows that there must be at least one $\to_{\text{od}}$-step in this $\to_{\text{oud}}$-rewrite sequence. Hence, by the construction of the rewrite sequence, we can conclude that at least one application of COMP or of c-COMP must be contained in the subderivation $\mathscr{D}_1$ of $\mathscr{D}$. □

The following theorem justifies the introduction of **e-BH$_0$** as a proof system with more inference rules than **BH$_0$** but the same set of theorems. The proof uses the fact that each application of c-FOLD$_{l/r}$ in a derivation $\mathscr{D}$ in **e-BH$_0$** can be eliminated individually by using the 'deductive power' of an application of COMP or c-COMP higher up in $\mathscr{D}$, which is guaranteed to exist by Lemma 5.8.

Theorem 5.9. Every derivation $\mathscr{D}$ in **e-BH$_0$** can be transformed into a derivation $\mathscr{D}'$ in **BH$_0$** with the same conclusion and the same (if any) open assumptions. As a consequence, **e-BH$_0$** has the same theorems as **BH$_0$**.

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

$$\dfrac{\dfrac{\dfrac{[\,s=\mu\beta.\,t]^u \quad\quad [\,s=\mu\beta.\,t]^u}{\mathcal{D}_{11} \quad\quad\quad \mathcal{D}_{1n}}}{\dfrac{s_{11}=t_{11} \quad\quad \ldots \quad\quad s_{1n}=t_{1n}}{F(s_{11},\ldots,s_{1n})=F(t_{11},\ldots,t_{1n})}\ \mathrm{COMP}}\ \mathrm{FOLD}_{l/r}}{\dfrac{s=t[\mu\beta.\,t/\beta]}{s=\mu\beta.\,t}\ \text{c-FOLD}_r,\ u}$$

Figure 16. The derivation $\mathscr{D}$ in **e-BH$_0$** that has an immediate subderivation $\mathscr{D}_1$ in **BH$_0$**.

$$\dfrac{\dfrac{\dfrac{\dfrac{(F(s_{11},\ldots,s_{1n})=F(t_{11},\ldots,t_{1n}))^u}{s=t[\mu\beta.\,t/\beta]}\ \mathrm{FOLD}_r}{[\,s=\mu\beta.\,t\,]}}{\mathcal{D}_{11}}\ \mathrm{FOLD}_{l/r} \quad \dfrac{\dfrac{\dfrac{(F(s_{11},\ldots,s_{1n})=F(t_{11},\ldots,t_{1n}))^u}{s=t[\mu\beta.\,t/\beta]}\ \mathrm{FOLD}_r}{[\,s=\mu\beta.\,t\,]}}{\mathcal{D}_{1n}}}{\dfrac{s_{11}=t_{11} \quad\quad \ldots \quad\quad s_{1n}=t_{1n}}{F(s_{11},\ldots,s_{1n})=F(t_{11},\ldots,t_{1n})}\ \text{c-COMP},\ u}\ \mathrm{FOLD}_{l/r}}{\dfrac{s=t[\mu\beta.\,t/\beta]}{s=\mu\beta.\,t}\ \mathrm{FOLD}_r}$$

Figure 17. The result $\mathscr{D}^{(1)}$ of transforming the **e-BH$_0$**-derivation $\mathscr{D}$ in Figure 17 into a **BH$_0$**-derivation with the same sets of open assumptions.

Sketch of proof. The second sentence of the theorem is implied by the first, since **e-BH$_0$** is an extension of **BH$_0$**, and, therefore, every derivation in **BH$_0$** is also a derivation in **e-BH$_0$** with the same (if any) open assumptions. For the first sentence, it suffices to show that topmost occurrences of the c-FOLD$_{l/r}$ rules in **e-BH$_0$**-derivations can always be eliminated. For this it is enough to establish: every derivation $\mathscr{D}$ in **e-BH$_0$** that has an application of one of the c-FOLD$_{l/r}$ rules at the bottom and contains no other application of either of these rules can effectively be transformed into a derivation $\mathscr{D}^{(1)}$ in **BH$_0$** with the same conclusion and the same open assumptions as $\mathscr{D}$.

Consider the case of a derivation $\mathscr{D}$ in **e-BH$_0$** with a bottommost application of c-FOLD$_r$ and an immediate subderivation in **BH$_0$** (the case with a bottommost application of c-FOLD$_l$ can be settled analogously). Then, by Lemma 5.8, $\mathscr{D}$ contains at least one application of COMP or c-COMP. We only consider the first case (the second case can be argued similarly), in which $\mathscr{D}$ is of the form shown in Figure 16, where the double lines represent a possibly empty sequence of applications of FOLD$_{l/r}$, and where there is at least one open occurrence of the marked formula $(s=\mu\beta.\,t)^u$ in one of $\mathscr{D}_{11},\ldots,\mathscr{D}_{1n}$ that gets discharged at the application of c-FOLD$_r$ at the bottom. It is then easy to see that $\mathscr{D}$ has the same conclusion and the same open assumption sets as the derivation $\mathscr{D}^{(1)}$ in **BH$_0$** that is shown in Figure 17. □

The following soundness and completeness result for **e-BH$_0$** is an immediate consequence of Theorems 5.9 and 5.3.

Corollary 5.10. **e-BH$_0$** is sound and complete with respect to $=_{\mathcal{T}}$.

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

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

$$\text{(REFL)} \quad g = g \qquad\qquad \text{(Assm)} \quad (g = h)^u \quad \text{(with } u \in Mk\text{)}$$

The *inference rules* of $\mathbf{BH}_{\mathbf{0}}^{\leftrightarrows}$:

$$\frac{\begin{array}{ccc}\mathcal{D}_1 & & \mathcal{D}_n\\ \langle\,\alpha_1 \,|\, \mathcal{E}_g\,\rangle = \langle\,\beta_1 \,|\, \mathcal{E}_h\,\rangle & \dots & \langle\,\alpha_n \,|\, \mathcal{E}_g\,\rangle = \langle\,\beta_n \,|\, \mathcal{E}_h\,\rangle\end{array}}{\langle\,\alpha \,|\, \underbrace{\{\alpha = F(\alpha_1,\dots,\alpha_n),\,\dots\}}_{=\mathcal{E}_g}\rangle = \langle\,\beta \,|\, \underbrace{\{\beta = F(\beta_1,\dots,\beta_n),\,\dots\}}_{=\mathcal{E}_h}\rangle}\ \text{COMP}$$

$$\frac{\begin{array}{ccc}\big[\,\langle\,\alpha \,|\, \mathcal{E}_g\,\rangle = \langle\,\beta \,|\, \mathcal{E}_h\,\rangle\,\big]^u & & \big[\,\langle\,\alpha \,|\, \mathcal{E}_g\,\rangle = \langle\,\beta \,|\, \mathcal{E}_h\,\rangle\,\big]^u\\ \mathcal{D}_1 & & \mathcal{D}_n\\ \langle\,\alpha_1 \,|\, \mathcal{E}_g\,\rangle = \langle\,\beta_1 \,|\, \mathcal{E}_h\,\rangle & \dots & \langle\,\alpha_n \,|\, \mathcal{E}_g\,\rangle = \langle\,\beta_n \,|\, \mathcal{E}_h\,\rangle\end{array}}{\langle\,\alpha \,|\, \underbrace{\{\alpha = F(\alpha_1,\dots,\alpha_n),\,\dots\}}_{=\mathcal{E}_g}\rangle = \langle\,\beta \,|\, \underbrace{\{\beta = F(\beta_1,\dots,\beta_n),\,\dots\}}_{=\mathcal{E}_h}\rangle}\ \begin{array}{l}\text{c-COMP, } u\\ \text{(if side-condition } \mathbf{I}\text{)}\end{array}$$

Figure 18. A Brandt–Henglein system $\mathbf{BH}_{\mathbf{0}}^{\leftrightarrows}$ without symmetry and transitivity rules for bisimulation equivalence between canonical term graph specifications.

6. Similar proof systems for equational term graph specifications

In this section, taking inspiration from the $\mathbf{BH_0}$ and $\mathbf{AK_0}$ variants of the **BH** and **AK** proof systems, we define a pair of proof systems that are concerned with bisimilarity on equational specifications of cyclic term graphs, as defined in Section 2.2.

First we define a proof system $\mathbf{BH}_{\mathbf{0}}^{\leftrightarrows}$ for the relation $\leftrightarrows$ on ctgs's that is analogous to the system $\mathbf{BH_0}$ for μ-terms defined in Section 5.

Definition 6.1. The natural-deduction style proof system $\mathbf{BH}_{\mathbf{0}}^{\leftrightarrows}$ has the equations between ctgs's as its *formulas*. It has the *axioms* (that is, *zero-premise rules*) REFL, allows the *assumptions* (Assm) and has the *inference rules* COMP and c-COMP shown in Figure 18. c-COMP is the single rule of $\mathbf{BH}_{\mathbf{0}}^{\leftrightarrows}$ at which assumptions can be, and some indeed must be, discharged: with reference to the designations used for the schematic application of this rule in Figure 18, the side-condition **I** stipulates that the set of open assumptions $(\langle\,\alpha \,|\, \mathscr{E}_g\,\rangle = \langle\,\beta \,|\, \mathscr{E}_h\,\rangle)^u$ in the derivations $\mathscr{D}_1, \dots, \mathscr{D}_n$ must be non-empty (these open assumptions are discharged by the application of c-COMP at the bottom).

A formula $g = h$ is a *theorem* of $\mathbf{BH}_{\mathbf{0}}^{\leftrightarrows}$, denoted $\vdash_{\mathbf{BH}_{\mathbf{0}}^{\leftrightarrows}} g = h$, if and only if there is a derivation in $\mathbf{BH}_{\mathbf{0}}^{\leftrightarrows}$ with conclusion $g = h$ and no open assumptions.

Theorem 6.2. The system $\mathbf{BH}_{\mathbf{0}}^{\leftrightarrows}$ is sound and complete with respect to bisimulation equivalence $\leftrightarrows$ on canonical term graph specifications: for all ctgs's g and h,

$$\vdash_{\mathbf{BH}_{\mathbf{0}}^{\leftrightarrows}} g = h \quad \Longleftrightarrow \quad g \leftrightarrows h \ . \tag{6.10}$$

Sketch of proof.

'⇒' The soundness direction is a consequence of the easily verifiable fact that every derivation $\mathscr{D}$ in $\mathbf{BH}_{\mathbf{0}}^{\leftrightarrows}$ with conclusion $g = h$ and no open assumptions corresponds directly to a bisimulation between the ctgs's g and h: it is easy to verify that $R = \{\langle lv(\tilde{g}), lv(\tilde{h})\rangle \mid \tilde{g} = \tilde{h} \text{ is a formula in } \mathscr{D}\}$ is a bisimulation between g and h.

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

The *inference rule* of $\mathbf{AK_0^{\leftrightarrows}}$:

$$\frac{\langle\, \alpha \,|\, \overbrace{\{\alpha = F(\alpha_1,\ldots,\alpha_n),\ldots\}}^{=\mathcal{E}_g}\,\rangle = \langle\, \beta \,|\, \overbrace{\{\beta = F(\beta_1,\ldots,\beta_n),\ldots\}}^{=\mathcal{E}_h}\,\rangle}{\langle\, \alpha_i \,|\, \mathcal{E}_g \,\rangle = \langle\, \beta_i \,|\, \mathcal{E}_h \,\rangle}\ \text{DECOMP}\ (\text{for } 1\leq i\leq n)$$

Figure 19. The Ariola–Klop system $\mathbf{AK_0^{\leftrightarrows}}$ for consistency checking with respect to bisimulation equivalence on canonical term graph specifications.

'$\Leftarrow$' The completeness direction can be proved by showing that, for all ctgs's g and h with $g \leftrightarrows h$, a straightforward bottom-up proof search from the equation $g = h$ in $\mathbf{BH_0^{\leftrightarrows}}$ is able to find a derivation in $\mathbf{BH_0^{\leftrightarrows}}$ with conclusion $g = h$ and no open assumptions. □

Next we introduce a proof system $\mathbf{AK_0^{\leftrightarrows}}$ for $\leftrightarrows$, which, as with $\mathbf{AK_0}$ for equivalence of μ-terms, is intended for consistency checking.

Definition 6.3. The Hilbert-style proof system $\mathbf{AK_0^{\leftrightarrows}}$ contains precisely all equations between ctgs's as its *formulas*. It contains *no axioms*. Its single *inference rule* is the rule DECOMP for which a schematic application is shown in Figure 19. For all g, h, g_1, h_1, we use $g = h \vdash_{\mathbf{AK_0^{\leftrightarrows}}} g_1 = h_1$ to denote the statement that there is a derivation in $\mathbf{AK_0^{\leftrightarrows}}$ from the assumption $g = h$ with conclusion $g_1 = h_1$.

In order to formulate a statement that links 'consistency with respect to **AK**' and bisimilarity of ctgs's, we need to define when a formula of $\mathbf{AK_0^{\leftrightarrows}}$ is to be called 'consistent' with respect to $\mathbf{AK_0^{\leftrightarrows}}$, and what we mean by 'contradictions with respect to $\leftrightarrows$'.

We say an equation $g = h$ between two ctgs's $g = \langle\, \alpha_0 \,|\, \{\alpha_0 = t_0, \ldots\}\,\rangle$ and $h = \langle\, \alpha'_0 \,|\, \{\alpha'_0 = t'_0, \ldots\}\,\rangle$ is a *contradiction with respect to* $\leftrightarrows$ if and only if the terms on the right-hand sides of the leading equations in g and h start with different symbols, that is, if and only if $t_0 \equiv F(\alpha_{01},\ldots,\alpha_{0n_0})$ and $t'_0 \equiv G(\alpha'_{01},\ldots,\alpha'_{0n'_0})$ for some $n_0, n'_0 \in \omega$, variables $\alpha_{01},\ldots,\alpha_{0n_0}$, $\alpha'_{01},\ldots,\alpha'_{0n'_0}$ and *different* symbols $F, G \in t$. We say an equation $g = h$ is $\mathbf{AK_0^{\leftrightarrows}}$*-consistent*, for ctgs's g and h, if and only if no contradiction with respect to $\leftrightarrows$ is derivable in $\mathbf{AK_0^{\leftrightarrows}}$ from $g = h$.

With these definitions, we have the following 'correspondence theorem' for $\mathbf{AK_0^{\leftrightarrows}}$.

Theorem 6.4. Consistency with respect to $\mathbf{AK_0^{\leftrightarrows}}$ corresponds to bisimilarity of canonical term graph specifications in the following sense: for all ctgs's g and h,

$$g = h \ \text{ is } \mathbf{AK_0^{\leftrightarrows}}\text{-consistent} \iff g \leftrightarrows h\ . \tag{6.11}$$

Figure 20 shows an example of a derivation in $\mathbf{BH_0^{\leftrightarrows}}$ without open assumptions that is related by mirroring to a successful consistency check in $\mathbf{AK_0^{\leftrightarrows}}$.

7. Duality results

In the main part of this section we develop a duality result between the systems **e-BH$_0$** and $\mathbf{AK_0}$, starting with two preparatory steps:

1 In Section 7.1 we define the concept of 'consistency unfolding in $\mathbf{AK_0}$' for a given equation as a formalisation of downwards-growing derivation trees that allow us to recognise the consistency in $\mathbf{AK_0}$ of the equation at the root.

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

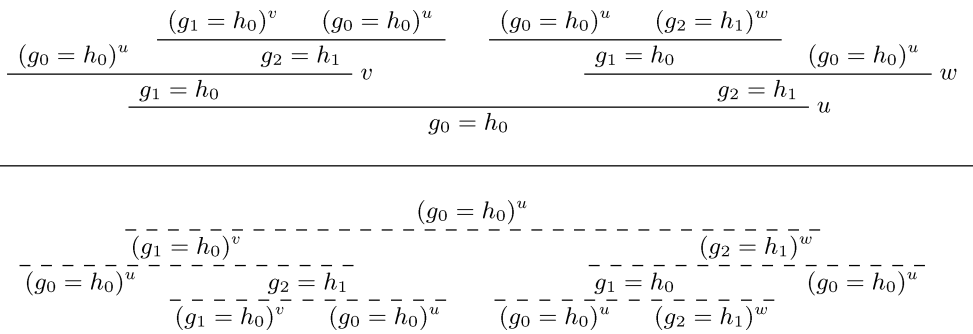

Figure 20. Example consisting of a derivation in $\mathbf{BH}_{\mathbf{0}}^{\leftrightarrows}$ without open assumptions and a successful consistency check in $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ that are close to each other's mirror image. For the bisimilar canonical term graph specifications $g = \langle \alpha_0 \,|\, \mathscr{E}_g \rangle$ and $h = \langle \beta_0 \,|\, \mathscr{E}_h \rangle$ from Example 2.17, here $g_i =_{\mathrm{def}} \langle \alpha_i \,|\, \mathscr{E}_g \rangle$ and $h_j =_{\mathrm{def}} \langle \beta_j \,|\, \mathscr{E}_h \rangle$, for all $i \in \{0,1,2\}$ and $j \in \{0,1\}$.

2 In Section 7.2 we define a pair of mirroring functions between consistency unfoldings in $\mathbf{AK_0}$ and derivations in $\mathbf{e\text{-}BH_0}$.

We then prove a duality theorem in Section 7.3 that states a correspondence by mirroring between consistency unfoldings in $\mathbf{AK_0}$ and derivations in $\mathbf{e\text{-}BH_0}$ without open assumptions.

In Section 7.4, we apply this theorem to give a new proof for the soundness of $\mathbf{e\text{-}BH_0}$ and $\mathbf{BH_0}$ with respect to $=_{\boldsymbol{T}}$. Finally, in Section 7.5, we indicate how the duality result between $\mathbf{BH_0}$ and $\mathbf{AK_0}$ can be transferred to a statement linking the proof systems $\mathbf{BH}_{\mathbf{0}}^{\leftrightarrows}$ and $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$ for equational specifications of cyclic term graphs.

7.1. *Consistency unfoldings in* $\mathbf{AK_0}$

In the first preparatory step we introduce 'consistency unfoldings' in $\mathbf{AK_0}$ for equations $s = t$ as downwards-growing derivation trees that allow us to recognise the consistency of $s = t$ with $\mathbf{AK_0}$ by similar arguments to those sketched in Section 1. Consistency unfoldings are built from rule applications in $\mathbf{AK_0}$ and from branchings that arise from the two possible conclusions of the decomposition rule. A consistency unfolding of an equation $s = t$ gives a finite overview of the derivations in $\mathbf{AK_0}$ that are possible from the assumption $s = t$ by gathering all such derivations until at some point looping or a reflexivity axiom is encountered. The derivations that 'span' a consistency unfolding are not required to be of minimal length to ensure looping or to have a reflexivity axiom as the conclusion.

Figure 21 provides a motivating example for the definition of the 'consistency unfolding' concept: it shows a derivation tree $\mathscr{C}$ that assembles, for μ-terms s and t that represent the bisimilar cyclic term graphs in Figure 3, all possible derivations from the assumption $s = t$ until looping occurs (for the first time). Single and double lines in $\mathscr{C}$ separate the respective premises and conclusions of one or two applications, respectively, of $\mathrm{UNFOLD}_{l/r}$, while branchings at dashed lines in $\mathscr{C}$ stem from the two possible ways in which conclusions can be deduced by applications of DECOMP. The markers u, v and w used for some formula

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

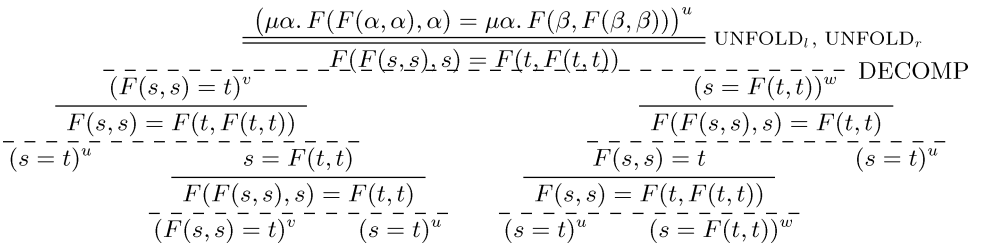

Figure 21. The consistency unfolding $\mathscr{C}$ in $\mathbf{AK_0}$ of $s = t$, where $s \equiv \mu\alpha.\, F(F(\alpha,\alpha),\alpha)$ and $t \equiv \mu\beta.\, F(\beta,F(\beta,\beta))$ are the μ-terms representing the term graphs $\mathscr{G}_2$ and $\mathscr{H}_2$ in Figure 3.

occurrences in $\mathscr{C}$ highlight the looping in those $\mathbf{AK_0}$-derivations initial segments of which constitute the branches of $\mathscr{C}$. Since $\mathscr{C}$ does not contain contradictions, $\mathscr{C}$ witnesses the consistency of $s = t$ with $\mathbf{AK_0}$ in the sense that its construction facilitates an easy proof by induction of this fact. The derivation tree $\mathscr{C}$ corresponds closely to the deduction graph in Figure 4, which demonstrates the consistency of an equation between equational specifications of the same cyclic term graphs (in a syntactic-matching system appropriate for dealing with such equational specifications).

We are going to define consistency unfoldings as downwards-growing derivation trees in which looping is described by markers that are attached to individual formulas: in a way that is reminiscent of derivations in natural-deduction systems without open assumptions, a leaf occurrence of a marked formula $(s = t)^u$ is bound to an 'internal occurrence' (the first encountered above) of the same marked formula inside the derivation tree. While in natural-deduction style derivations without open assumptions the assumptions are bound to applications of rules, in consistency unfoldings, leaf occurrences of marked formulas are 'bound back' to occurrences of the same formula higher up. The use of markers to express these bindings is significant for our later results, because it provides a match with derivations in $\mathbf{e\text{-}BH_0}$ in which assumption markers indicate the structure of the bindings of individual assumptions to individual rule applications.

In order to define consistency unfoldings, we need the concept of 'partial consistency unfolding', in which, analogously to derivations with open assumptions, not all leaf occurrences of marked formulas have to be 'bound back'. We again assume a countably infinite set *Mk* of *markers* for the following definition.

Definition 7.1. A *partial consistency unfolding* (a *pcu*) in $\mathbf{AK_0}$ of an equation $s = t$, where $s, t \in T_\mu(\Sigma)$, is a downwards-growing prooftree that can be formed by a finite number of applications of the six generating clauses (i)–(vi) in Figure 22 (the notation used in the clauses illustrated in Figure 22 is explained below). This inductive definition simultaneously defines both the *depth* $|\mathscr{C}|$ and the set $\mathsf{ucons}(\mathscr{C})$ of *unbound consequences* in $\mathscr{C}$ (of the formula $s = t$ with respect to $\mathbf{AK_0}$).

In the generating clauses in Figure 22, both the pcu's that are assumed and the pcu that is generated by a clause have been put into framed boxes that are *not* part of the defined objects. Each of the clauses (i)–(vi) should be read as follows: if the zero, one, or one, ..., n pcu's (denoted by $\mathscr{C}_i$ for $i \in \{1,\ldots,n\}$) of the form as described on the left-hand side

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

Clause	*pcu's $\mathcal{C}_i$ assumed*	*pcu $\mathcal{C}$ generated by the clause*
(i)	—	$\boxed{t = t}$ $\mathsf{ucons}(\mathcal{C}) =_{\mathrm{def}} \emptyset$ and $\lvert\mathcal{C}\rvert =_{\mathrm{def}} 0$.
(ii)	—	$\boxed{(s = t)^u}$ $\mathsf{ucons}(\mathcal{C}) =_{\mathrm{def}} \{(s = t)^u\}$ and $\lvert\mathcal{C}\rvert =_{\mathrm{def}} 0$.
(iii)	$\boxed{\begin{array}{c}(s_1 = t_1)^{\boldsymbol{m}_1}\\ \mathcal{C}_1\end{array}}$	$\boxed{\begin{array}{c}\dfrac{s = t}{(s_1 = t_1)^{\boldsymbol{m}_1}}\,R\\ \mathcal{C}_1\end{array}}$ where $R \in \{\mathrm{UNFOLD}_l, \mathrm{UNFOLD}_r\}$, $\mathsf{ucons}(\mathcal{C}) =_{\mathrm{def}} \mathsf{ucons}(\mathcal{C}_1)$, and $\lvert\mathcal{C}\rvert =_{\mathrm{def}} \lvert\mathcal{C}_1\rvert + 1$.
(iv)	$\boxed{\begin{array}{c}(s_1 = t_1)^{\boldsymbol{m}_1}\\ \mathcal{C}_1\\ [s = t]^u\end{array}}$ with $(s = t)^u \in \mathsf{ucons}(\mathcal{C}_1)$;	$\boxed{\begin{array}{c}\dfrac{(s = t)^u}{(s_1 = t_1)^{\boldsymbol{m}_1}}\,R\\ \mathcal{C}_1\\ [s = t]^u\end{array}}$ where $R \in \{\mathrm{UNFOLD}_l, \mathrm{UNFOLD}_r\}$, $\mathsf{ucons}(\mathcal{C}) =_{\mathrm{def}} \mathsf{ucons}(\mathcal{C}_1) \setminus \{(s = t)^u\}$, and $\lvert\mathcal{C}\rvert =_{\mathrm{def}} \lvert\mathcal{C}_1\rvert + 1$.
(v)	$\boxed{\begin{array}{c}(s_i = t_i)^{\boldsymbol{m}_i}\\ \mathcal{C}_i\end{array}}$ for all $i \in \{1, \ldots, n\}$	$\boxed{\begin{array}{c}\dfrac{F(s_1, \ldots, s_n) = F(t_1, \ldots, t_n)}{(s_1 = t_1)^{\boldsymbol{m}_1} \;\cdots\; (s_n = t_n)^{\boldsymbol{m}_n}}\,\mathrm{DECOMP}\\ \mathcal{C}_1 \qquad\qquad \mathcal{C}_n\end{array}}$ with a DECOMP-*branching* at the top, $\mathsf{ucons}(\mathcal{C}) =_{\mathrm{def}} \mathsf{ucons}(\mathcal{C}_1) \cup \ldots \cup \mathsf{ucons}(\mathcal{C}_n)$, and $\lvert\mathcal{C}\rvert =_{\mathrm{def}} 1 + \max\{\lvert\mathcal{C}_1\rvert, \ldots, \lvert\mathcal{C}_n\rvert\}$.
(vi)	$\boxed{\begin{array}{c}(s_i = t_i)^{\boldsymbol{m}_i}\\ \mathcal{C}_i\\ [s = t]^u\end{array}}$ for all $i \in \{1, \ldots, n\}$, where $(s = t)^u \in \bigcup_{i=1}^n \mathsf{ucons}(\mathcal{C}_i)$ with $s \equiv F(s_1, \ldots, s_n)$ and $t \equiv F(t_1, \ldots, t_n)$;	$\boxed{\begin{array}{c}\dfrac{(F(s_1, \ldots, s_n) = F(t_1, \ldots, t_n))^u}{(s_1 = t_1)^{\boldsymbol{m}_1} \;\cdots\; (s_n = t_n)^{\boldsymbol{m}_n}}\,\mathrm{DECOMP}\\ \mathcal{C}_1 \qquad\qquad \mathcal{C}_n\\ [s = t]^u \qquad\qquad [s = t]^u\end{array}}$ with a DECOMP-*branching* at the top, $\mathsf{ucons}(\mathcal{C}) =_{\mathrm{def}} \left(\bigcup_{i=1}^n \mathsf{ucons}(\mathcal{C}_i)\right) \setminus \{(s = t)^u\}$, and $\lvert\mathcal{C}\rvert =_{\mathrm{def}} 1 + \max\{\lvert\mathcal{C}_1\rvert, \ldots, \lvert\mathcal{C}_n\rvert\}$.

Figure 22. The six generating clauses of partial consistency unfoldings in $\mathbf{AK_0}$.

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

of the clause are assumed, then the derivation tree on the right-hand side of Figure 22 is a partial consistency unfolding of the new formula at its top. One further restriction not mentioned in Figure 22 is required for the generating steps (v) and (vi): different formulas with the same markers attached must not occur in the union of the sets $\mathsf{ucons}(\mathscr{C}_1), \ldots, \mathsf{ucons}(\mathscr{C}_n)$ for the pcu's $\mathscr{C}_1, \ldots, \mathscr{C}_n$ assumed.

All formula occurrences in a pcu $\mathscr{C}$ of $s = t$ are called *consequences* of $s = t$ in $\mathscr{C}$. Markers attached to formula occurrences at the bottom (*leaf occurrences*) and to formula occurrences at internal nodes (*internal occurrences*) of a pcu $\mathscr{C}$ denote bindings between leaf occurrences and internal occurrences higher up in $\mathscr{C}$ of the same marked formula: a leaf occurrence of $(\tilde{s} = \tilde{t})^u$ at the bottom of $\mathscr{C}$ is said to be *bound back* to the nearest internal occurrence of $(\tilde{s} = \tilde{t})^u$ higher up in $\mathscr{C}$. If for a leaf occurrence of $(\tilde{s} = \tilde{t})^u$ there does not exist an internal occurrence higher up of the same marked formula, then the leaf occurrence is called an *unbound consequence* in $\mathscr{C}$; in this case the definition of pcu's warrants $(\tilde{s} = \tilde{t})^u \in \mathsf{ucons}(\mathscr{C})$.

The expression $[s = t]^u$ at the bottom of $\mathscr{C}_1$ (respectively, of $\mathscr{C}_1, \ldots, \mathscr{C}_n$) on the left in clause (iv) (respectively, in clause (vi)) denotes the set of unbound consequences in $\mathscr{C}_1$ (respectively, in $\mathscr{C}_1, \ldots, \mathscr{C}_n$) of the form $(s = t)^u$. And then the expression $[s = t]^u$ on the right-hand side in clause (iv) (respectively, in clause (vi)), stands for a corresponding set of consequences in the generated pcu $\mathscr{C}$ that in $\mathscr{C}$ are now bound back to the new occurrence of the marked formula $(s = t)^u$ at the top (respectively, the root of $\mathscr{C}$).

In clauses (i)–(vi), and also later, marked formulas $(s = t)^{\boldsymbol{m}}$ with $s, t \in T_\mu(\Sigma)$ and a boldface-marker $\boldsymbol{m}$ stand for either:

(a) the unmarked formula $s = t$; or
(b) a marked formula $(s = t)^u$ with some $u \in Mk$, which in this case is denoted by $\boldsymbol{m}$.

For parts in a pcu $\mathscr{C}$ in $\mathbf{AK_0}$ that involve a dashed line and are of the form

$$\frac{(F(s_1, \ldots, s_n) = F(t_1, \ldots, t_n)^{\boldsymbol{m}}}{(s_1 = t_1)^{\boldsymbol{m}_1} \quad \cdots \quad (s_n = t_n)^{\boldsymbol{m}_2}} \text{ DECOMP}$$

(which are produced in the course of generation steps (v) and (vi)), we will use the term *branching*. For such a branching we say that the formulas $F(s_1, \ldots, s_n) = F(t_1, \ldots, t_n)$, $s_1 = t_1, \ldots, s_n = t_n$ are its *premise*, *first conclusion*, ..., and *n-th conclusion*, respectively.

A partial consistency unfolding in $\mathbf{AK_0}$ is then defined to be a pcu of $s = t$ in $\mathbf{AK_0}$, for some $s, t \in T_\mu(\Sigma)$. We will use the letter $\mathscr{C}$, possibly with sub- and/or superscripts, as a syntactical variable that ranges over a partial consistency unfoldings.

Definition 7.2. Let $s, t \in T_\mu(\Sigma)$. A partial consistency unfolding $\mathscr{C}$ of $s = t$ in $\mathbf{AK_0}$ is called a *consistency unfolding* (a *cu*) of $s = t$ in $\mathbf{AK_0}$ if and only if $\mathsf{ucons}(\mathscr{C}) = \varnothing$ (that is, if and only if $\mathscr{C}$ does not contain unbound consequences of $s = t$). A cu of $s = t$ in $\mathbf{AK_0}$, for some $s, t \in T_\mu(\Sigma)$ is said to be a consistency unfolding in $\mathbf{AK_0}$.

Example 7.3. According to Definition 7.2, the derivation tree $\mathscr{C}$ depicted in Figure 21 can now be recognised as a pcu in $\mathbf{AK_0}$ without unbound consequences, and hence as a consistency unfolding of $\mu\alpha.\, F(F(\alpha, \alpha), \alpha) = \mu\beta.\, F(\beta, F(\beta, \beta))$.

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

Having formally introduced consistency unfoldings in $\mathbf{AK_0}$ and proposed them as adequate formalisations of successful consistency checks with respect to $\mathbf{AK_0}$ by way of Example 7.3, we now have to ask whether consistency unfoldings in $\mathbf{AK_0}$ do indeed 'witness' the consistency with respect to $\mathbf{AK_0}$ of the equation at their root.

The lemma below answers this question in the affirmative. To prove it, we use the fact that, analogously to derivations in $\mathbf{BH_0}$ without open assumptions, consistency unfoldings in $\mathbf{AK_0}$ correspond to finite bisimulations on $T_\mu(\Sigma)$ (up to adding $\to_{\mathrm{ou}}$-reachable pairs), and hence demonstrate, by the finite bisimulation principle, the equivalence of the μ-terms at the root.

Lemma 7.4. For all $s, t \in T_\mu(\Sigma)$,

$$(\exists \mathscr{C}) \begin{bmatrix} \mathscr{C} \text{ is a consistency unfolding} \\ \text{of } s = t \text{ in } \mathbf{AK_0} \end{bmatrix} \implies s = t \text{ is } \mathbf{AK_0}\text{-consistent} . \tag{7.12}$$

Sketch of proof. Let $\mathscr{C}$ be a consistency unfolding of $s = t$ in $\mathbf{AK_0}$, for some $s, t \in T_\mu(\Sigma)$. We can use an argument analogous to that used in the proof sketch for Theorem 5.3 to show that the relation $R =_{\mathrm{def}} \{\langle \tilde{s}, \tilde{t}\rangle \mid \tilde{s} = \tilde{t} \text{ occurs in } \mathscr{C}\}$ is a finite bisimulation up to adding $\to_{\mathrm{ou}}$-reachable pairs that includes the pair $\langle s, t\rangle$. Hence the extension $\tilde{R}$ of R defined as in (2.1) is a finite bisimulation on $T_\mu(\Sigma)$ with $\langle s, t\rangle \in \tilde{R}$. Now, the finite bisimulation principle, Theorem 2.13, gives $s =_{\boldsymbol{T}} t$. Finally, by Theorem 5.6, it follows that $s = t$ is consistent with respect to $\mathbf{AK_0}$. □

Lemma 7.4 guarantees that a consistency unfolding in $\mathbf{AK_0}$ of an equation $s = t$ witnesses the consistency of $s = t$ with $\mathbf{AK_0}$. But it leaves open the question as to whether the concept of consistency unfolding is indeed general enough to capture entirely the consistency of formulas with respect to $\mathbf{AK_0}$. The following theorem states that this is indeed the case.

Theorem 7.5. For all $s, t \in T_\mu(\Sigma)$,

$$(\exists \mathscr{C}) \begin{bmatrix} \mathscr{C} \text{ is a consistency unfolding} \\ \text{of } s = t \text{ in } \mathbf{AK_0} \end{bmatrix} \iff s = t \text{ is } \mathbf{AK_0}\text{-consistent.} \tag{7.13}$$

Sketch of proof. Let $s, t \in T_\mu(\Sigma)$.

'$\Rightarrow$' This is an instance of Lemma 7.4.

'$\Leftarrow$' This can be shown by the following argument, which is analogous to, in fact, as good as 'dual' to, the one used in a proof following Brandt and Henglein for the completeness of **BH** with respect to $=_{\boldsymbol{T}}$.

Suppose that $s =_{\boldsymbol{T}} t$. Then a consistency unfolding of $s = t$ in $\mathbf{AK_0}$ can be reached by building up the downwards-growing 'tree of consequences' of this equation in $\mathbf{AK_0}$ in successive extension stages, always stopping to expand a branch any further as soon as 'looping' occurs (that is, whenever a formula $s' = t'$ is produced of which another occurrence higher up on the same branch has already been passed) or as soon as a formula $r = r$ is encountered. The fact that there can be no infinite branches in the

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

resulting derivation tree is an easy consequence of the following three facts:

(a) Branches in the derivation tree correspond to $\mathbf{AK_0}$-derivations.
(b) Because of Proposition 5.5, the subformula principle, for all formulas $s_1 = t_1$ in the derivation tree, s_1 and t_1 are generated subterms of s and t, respectively.
(c) The sets $GS(s)$ and $GS(t)$ of generated subterms of s and t are finite because of Lemma 2.6. □

7.2. *Mirroring functions*

In the second preparatory step we give a definition of a pair of mirroring functions $\mathscr{D}$ and $\mathscr{C}$ between partial consistency unfoldings in $\mathbf{AK_0}$ and derivations in $\mathbf{e\text{-}BH_0}$.

The possibility of defining such mirroring functions rests on the fact that there exists a duality between $\mathbf{BH_0}$ and $\mathbf{AK_0}$ on the level of their rules: every one-premise rule R of $\mathbf{BH_0}$ corresponds to a rule R' of $\mathbf{AK_0}$ other than DECOMP such that every instance of R can be related to an instance of R' by swapping the roles of premise and conclusion. For example, every instance of the rule FOLD_l in $\mathbf{BH_0}$ corresponds to an instance of the rule UNFOLD_l in $\mathbf{AK_0}$ in the sense that, for all $s, t \in T_\mu(\Sigma)$, $\alpha \in Var$ and derivations $\mathscr{D}_1$ in $\mathbf{BH_0}$, the application at the bottom of the derivation

$$\mathrm{FOLD}_l\ \frac{\begin{array}{c}\mathscr{D}_1\\ s[\mu\alpha.\, s/\alpha] = t\end{array}}{\mu\alpha.\, s = t}$$

corresponds to

$$\frac{\mu\alpha.\, s = t}{s[\mu\alpha.\, s/\alpha] = t}\ \mathrm{UNFOLD}_l$$

There is also an obvious correspondence between instances of the composition rule COMP in $\mathbf{BH_0}$ and the decomposition branchings DECOMP in partial consistency unfoldings in $\mathbf{AK_0}$. That is, for all $s_1, t_1, \ldots, s_n, t_n \in T_\mu(\Sigma)$, the inference figures

$$\mathrm{COMP}\ \frac{s_1 = t_1 \quad \ldots \quad s_n = t_n}{F(s_1, \ldots, s_n) = F(t_1, \ldots, t_n)} \qquad \frac{F(s_1, \ldots, s_n) = F(t_1, \ldots, t_n)}{s_1 = t_1 \quad \ldots \quad s_n = t_n}\ \mathrm{DECOMP}$$

correspond by exchanging the roles of premises and conclusions.

In order to use this duality in the definition of transformations between prooftrees and derivation trees, we introduce the following notation.

Notation 7.6. We use the syntactical variables $R^{(d)}$ and $R^{(cu)}$ for rules in $\mathbf{BH_0}$ (and hence also in $\mathbf{e\text{-}BH_0}$) and $\mathbf{AK_0}$, respectively, to denote the bijective relationship between one-premise rules in $\mathbf{BH_0}$ and rules in $\mathbf{AK_0}$ different from DECOMP that is described by Table 1. The intended use of $R^{(d)}$ and $R^{(cu)}$ is explained by the following example: if in a particular context (of a proof, an argument, and so on) the syntactical variable $R^{(d)}$ is used for the FOLD_r rule in $\mathbf{BH_0}$ (or in $\mathbf{e\text{-}BH_0}$), then, in the same context, $R^{(cu)}$ will stand for the UNFOLD_r rule in $\mathbf{AK_0}$.

We now give the definitions of 'mirroring functions' $\mathscr{D}$ and $\mathscr{C}$ that, as will be shown in the next section, map pcu's in $\mathbf{AK_0}$ to derivations in $\mathbf{e\text{-}BH_0}$, and derivations in $\mathbf{e\text{-}BH_0}$ to pcu's in $\mathbf{AK_0}$, respectively.

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

Table 1. *Notation formalising the duality between rules of* $\mathbf{BH_0}$ *and rules of* $\mathbf{AK_0}$

Rule $R^{(d)}$ in $\mathbf{BH_0}$	FOLD_l	FOLD_r
Rule $R^{(cu)}$ in $\mathbf{AK_0}$	UNFOLD_l	UNFOLD_r

Definition 7.7. In item (i) we define a *mirroring function* $\mathscr{D}$, that maps partial consistency unfoldings in $\mathbf{AK_0}$ into prooftrees labelled by rules of $\mathbf{e\text{-}BH_0}$; in item (ii) we define a *mirroring function* $\mathscr{C}$ that maps derivations in $\mathbf{e\text{-}BH_0}$ into derivation trees containing marked formulas and involving $\mathbf{AK_0}$-rules and DECOMP-branchings.

(i) For every partial consistency unfolding $\mathscr{C}$ of $s = t$ in $\mathbf{AK_0}$, where $s, t \in T_\mu(\Sigma)$, the *mirroring* $\mathscr{D}(\mathscr{C})$ of $\mathscr{C}$ is defined by induction on the depth $|\mathscr{C}|$ of $\mathscr{C}$ according to the six clauses shown in Figure 23 (through the arrows $\overset{\mathscr{D}}{\longmapsto}$ from left to right) that arise by case-distinction dependent on the last step of the generation of $\mathscr{C}$ according to Definition 7.1. The first, third and fifth inductive clauses apply to cases of partial consistency unfoldings $\mathscr{C}$ where the formula at the root of $\mathscr{C}$ is not marked; the second, fourth and sixth clauses apply to cases where it is marked.

(ii) For every derivation $\mathscr{D}$ in $\mathbf{e\text{-}BH_0}$ with conclusion $s = t$, where $s, t \in T_\mu(\Sigma)$, the *mirroring* $\mathscr{C}(\mathscr{D})$ of $\mathscr{D}$ is defined by induction on the depth $|\mathscr{D}|$ of $\mathscr{D}$ according to the six inductive clauses (indicated by the arrows $\overset{\mathscr{C}}{\longmapsto}$ from right to left) in Figure 23. These six clauses cover all cases of axioms, assumptions and last rule applications in $\mathbf{e\text{-}BH_0}$-derivations.

7.3. *The duality result linking* $\mathbf{e\text{-}BH_0}$ *and* $\mathbf{AK_0}$

The lemma below is central to our duality result. On the one hand, it justifies the well-definedness of the mirroring functions $\mathscr{D}$ and $\mathscr{C}$ in the sense that the images of $\mathscr{D}$ and $\mathscr{C}$ are derivations in $\mathbf{e\text{-}BH_0}$ (with possibly open assumptions) and pcu's in $\mathbf{AK_0}$, respectively. On the other hand, it will lead us directly to our main result.

Lemma 7.8. The mirroring functions $\mathscr{C}$ and $\mathscr{D}$ define a bijective relationship between partial consistency unfoldings in $\mathbf{AK_0}$ and derivations in $\mathbf{e\text{-}BH_0}$ in the sense explained in items (i), (ii) and (iii) below and illustrated by the following picture:

$$\mathscr{C}(\mathscr{D}) = \boxed{\begin{array}{c}(s=t)^{\boldsymbol{m}}\\ \mathscr{C}\\ \{\,[s_i=t_i]^{u_i}\,\}_{i=1,\ldots,n}\end{array}} \quad \begin{array}{c}\overset{\mathscr{D}}{\longmapsto}\\ \underset{\mathscr{C}}{\longmapsfrom}\end{array} \quad \boxed{\begin{array}{c}\{\,[s_i=t_i]^{u_i}\,\}_{i=1,\ldots,n}\\ \mathscr{D}\\ s=t\end{array}} = \mathscr{D}(\mathscr{C}) \qquad (7.14)$$

(at the bottom of $\mathscr{C}$ the family $\{\,[s_i = t_i]^{u_i}\,\}_{i=1,\ldots,n}$ denotes the sets of unbound consequences in $\mathscr{C}$, at the top of $\mathscr{D}$ this family symbolises the open assumption sets of $\mathscr{D}$).

(i) Let $\mathscr{C}$ be a partial consistency unfolding of $s = t$ in $\mathbf{AK_0}$, for some $s, t \in T_\mu(\Sigma)$. Then the mirroring $\mathscr{D}(\mathscr{C})$ of $\mathscr{C}$ is a derivation in $\mathbf{e\text{-}BH_0}$ with conclusion $s = t$ and with the same sets of open assumptions as $\mathscr{C}$ has sets of unbound consequences.

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

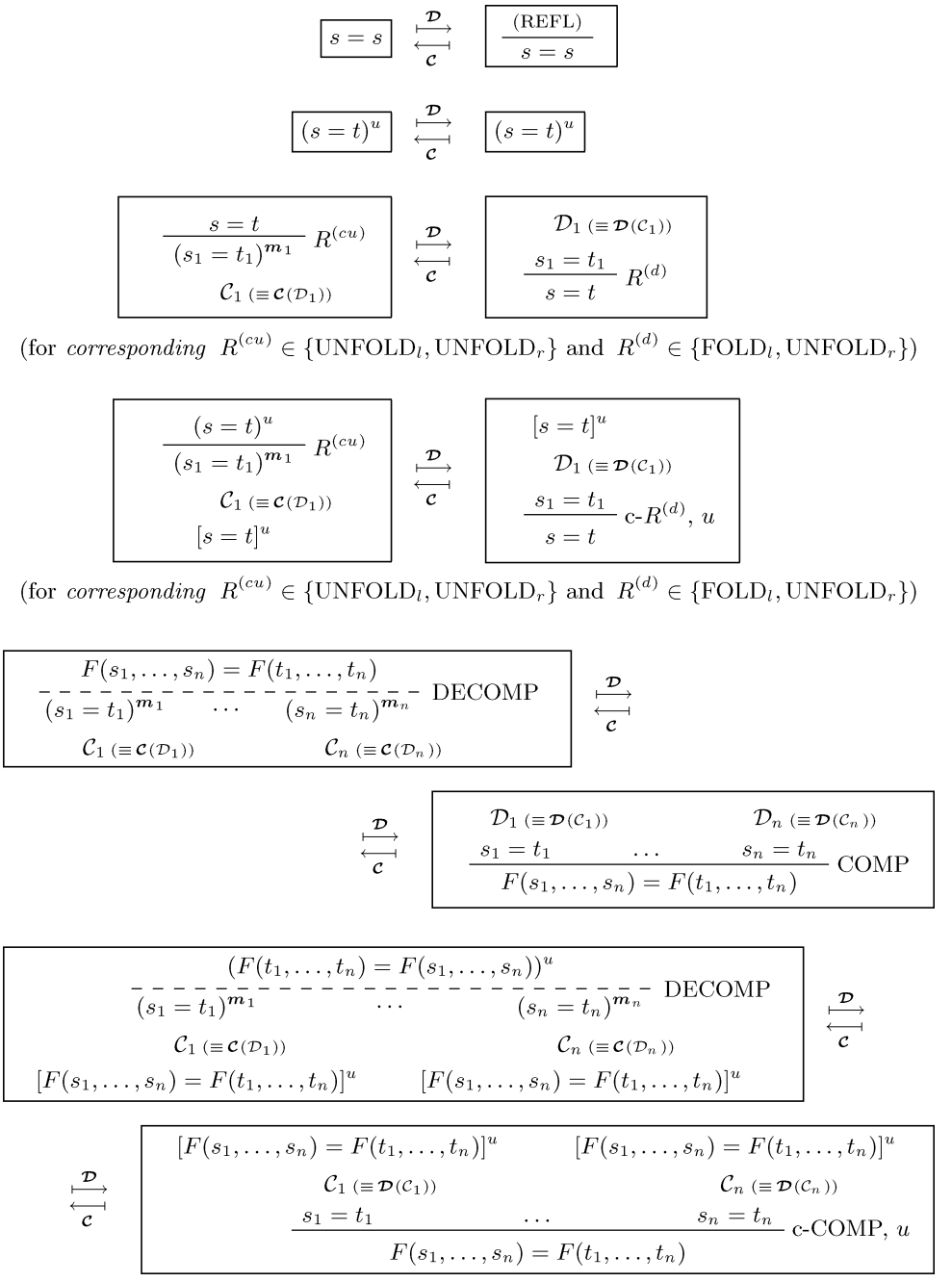

Figure 23. The six clauses of the inductive definitions of the mirroring functions $\mathscr{D}$ and $\mathscr{C}$: the function $\mathscr{D}$ maps partial consistency unfoldings $\mathscr{C}$ in $\mathbf{AK_0}$ into derivations $\mathscr{D}(\mathscr{C})$ in $\mathbf{e\text{-}BH_0}$, and the function $\mathscr{C}$ maps derivations $\mathscr{D}$ in $\mathbf{e\text{-}BH_0}$ into partial consistency unfoldings $\mathscr{C}(\mathscr{D})$ in $\mathbf{AK_0}$.

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

(ii) Let $\mathscr{D}$ be a derivation in **e-BH$_0$** with conclusion $s = t$, for some $s, t \in T_\mu(\Sigma)$. Then the mirroring $\mathscr{C}(\mathscr{D})$ of $\mathscr{D}$ is a partial consistency unfolding of $s = t$ in **AK$_0$** that has the same sets of unbound consequences as $\mathscr{D}$ has open assumption sets.

(iii) The mirroring functions $\mathscr{D}$ and $\mathscr{C}$ are inverses of each other.

Proof.

(i) We use induction on the depth $|\mathscr{C}|$ of consistency unfoldings $\mathscr{C}$ in **AK$_0$**.

The base case of a pcu $\mathscr{C}$ with $|\mathscr{C}| = 0$ is settled by noticing that an unmarked formula $s = s$ and a marked formula $(s = t)^u$ are mapped by the mirroring function $\mathscr{D}$ to an axiom (REFL) and to the assumption $(s = t)^u$, respectively.

For the induction step, we will only consider the case of a partial consistency unfolding $\mathscr{C}$ with $|\mathscr{C}| \geqslant 1$ for which the inductive definition of $\mathscr{D}(\mathscr{C})$ in the outermost induction step proceeds by an application of the sixth clause in Figure 23. The argument in the other five cases is either much easier or, in the case of a required application of the fourth clause in Figure 23, involves a very similar argument to the one given in the case considered in (ii) below.

Let $\mathscr{C}$ be an arbitrary partial consistency unfolding in **AK$_0$** of the form for some $\tilde{s}_1, \ldots, \tilde{s}_n, \tilde{t}_1, \ldots, \tilde{t}_n \in Ter_\mu(\Sigma)$ and a marker u, where the sets of unbound consequences are precisely those that belong to the family $\{[s_i = t_i]^{u_i}\}_{i=1,\ldots,n}$ for some $n \in \omega$ and $s_i, t_i \in Ter_\mu(\Sigma)$, and a marker u_i for $i = 1, \ldots, n$. In this way, the respective parts in $\mathscr{C}_1$, $\ldots$, $\mathscr{C}_n$ of the unbound consequences in $\mathscr{C}$ that belong to one of these sets $[s_i = t_i]^{u_i}$ are gathered at the bottom of $\mathscr{C}_1, \ldots, \mathscr{C}_n$ by the respective families, each of which is denoted there by $\{[s_i = t_i]^{u_i}\}_{i=1,\ldots,n}$.

Furthermore, $[F(\tilde{s}_1, \ldots, \tilde{s}_n) = F(\tilde{t}_1, \ldots, \tilde{t}_n)]^u$ at the bottom of $\mathscr{C}_1$, and $\mathscr{C}_n$ denotes the set of all such leaf occurrences of $(F(\tilde{s}_1, \ldots, \tilde{s}_n) = F(\tilde{t}_1, \ldots, \tilde{t}_n))^u$ in $\mathscr{C}_1, \ldots, \mathscr{C}_n$ that are not bound back in $\mathscr{C}_1, \ldots, \mathscr{C}_n$, but are bound back in $\mathscr{C}$ to the root formula. Since $\mathscr{C}$ is a pcu that must have been formed in the last generating step according to clause (vi) in Definition 7.1, at least one such leaf occurrence of $(F(\tilde{s}_1, \ldots, \tilde{s}_n) = F(\tilde{t}_1, \ldots, \tilde{t}_n)^u)$ that is bound back to the root of $\mathscr{C}$ occurs in one of the parts $\mathscr{C}_1, \ldots, \mathscr{C}_n$ of $\mathscr{C}$.

By the induction hypothesis, we find that $\mathscr{D}(\mathscr{C}_1), \ldots, \mathscr{D}(\mathscr{C}_n)$ are **e-BH$_0$**-derivations with conclusions $\tilde{s}_1 = \tilde{t}_1, \ldots, \tilde{s}_n = \tilde{t}_n$, which together possess precisely the sets $[F(\tilde{s}_1, \ldots, \tilde{s}_n) = F(\tilde{t}_1, \ldots, \tilde{t}_n)]^u$ and $[s_i = t_i]^{u_i}$ for $i \in \{1, \ldots, n\}$ as open assumption sets that are non-empty in at least one of $\mathscr{D}(\mathscr{C}_1), \ldots, \mathscr{D}(\mathscr{C}_n)$. From this it follows that the result $\mathscr{D}(\mathscr{C})$

$$\dfrac{\begin{array}{llll} [F(\tilde{s}_1, \ldots, \tilde{s}_n) = F(\tilde{t}_1, \ldots, \tilde{t}_n)]^u & & & [F(\tilde{s}_1, \ldots, \tilde{s}_n) = F(\tilde{t}_1, \ldots, \tilde{t}_n)]^u \\ \{[s_i = t_i]^{u_i}\}_{i=1,\ldots,n} & & & \{[s_i = t_i]^{u_i}\}_{i=1,\ldots,n} \\ \mathscr{D}(\mathscr{C}_1) & & & \mathscr{D}(\mathscr{C}_n) \\ \tilde{s}_1 = \tilde{t}_1 & \ldots & & \tilde{s}_n = \tilde{t}_n \end{array}}{F(\tilde{s}_1, \ldots, \tilde{s}_n) = F(\tilde{t}_1, \ldots, \tilde{t}_n)}\ \text{c-COMP}, u$$

of applying the mirroring function $\mathscr{D}$ to $\mathscr{C}$ is an **e-BH$_0$**-derivation with conclusion $F(\tilde{s}_1, \ldots, \tilde{s}_n) = F(\tilde{t}_1, \ldots, \tilde{t}_n)$ whose open assumption sets are precisely those of the family $\{[s_i = t_i]^{u_i}]_{i=1,\ldots,n}$ (in particular, we conclude here that the side-condition **I** on

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

the application of c-COMP at the bottom of $\mathscr{D}(\mathscr{C})$ is fulfilled). Hence we have shown the statement required for the induction step in this case.

(ii) Analogously to the proof of item (i), we carry out the proof of the statement in item (ii) of the lemma by induction on the depth $|\mathscr{D}|$ of **e-BH$_0$**-derivations $\mathscr{D}$.
The base case of the induction is straightforward.
For the induction step, we will only consider the case of an **e-BH$_0$**-derivation $\mathscr{D}$ with $|\mathscr{D}| \geqslant 1$ in which the last rule application in $\mathscr{D}$ is a rule c-$R^{(d)}$ for some rule $R^{(d)} \in \{\mathrm{FOLD}_l, \mathrm{FOLD}_r\}$. The cases of **e-BH$_0$**-derivations $\mathscr{D}$ with other rules applied at the bottom can be treated in a similar and rather easier way (the case with c-COMP as last rule application in $\mathscr{D}$ can be established by 'reversing' the reasoning in the case detailed for the induction step in (i)).
Hence, we assume that $\mathscr{D}$ is an **e-BH$_0$**-derivation of the form

$$\dfrac{\begin{array}{c}[s=t]^u \quad \{[s_i=t_i]^{u_i}\}_{i=1,\ldots,n} \\ \mathscr{D}_1 \\ \tilde{s}_1=\tilde{t}_1\end{array}}{s=t}\ \text{c-}R^{(d)},\ u$$

for an $n \in \omega$ and some $s, t, \tilde{s}_1, \tilde{t}_1, s_1, t_1, \ldots, s_n, t_n \in Ter_\mu(\Sigma)$, as well as with sets $[s_i = t_i]^{u_i}$ of open assumptions, for $i \in \{1, \ldots, n\}$. We notice that, due to the side-condition **I** on the application of c-$R^{(d)}$ at the bottom of $\mathscr{D}$, the open assumption set $[s=t]^u$ in $\mathscr{D}_1$, which consists of the assumptions that are discharged at the bottommost rule application in $\mathscr{D}$, must be non-empty. By applying the induction hypothesis for $\mathscr{D}_1$, we find that $\mathscr{C}(\mathscr{D}_1)$ is a pcu of $\tilde{s}_1 = \tilde{t}_1$ in **AK$_0$** that contains exactly the sets $[s=t]^u$ and $[s_i=t_i]^{u_i}$, for $i \in \{1, \ldots, n\}$, of unbound consequences at the bottom of $\mathscr{C}(\mathscr{D}_1)$. With this, it follows that the derivation tree $\mathscr{C}(\mathscr{D})$

$$\dfrac{\begin{array}{c}\dfrac{(s=t)^u}{(\tilde{s}_1=\tilde{t}_1)^{\boldsymbol{m}_1}}\ R^{(cu)} \\ \mathscr{C}(\mathscr{D}_1) \\ [s=t]^u \quad \{[s_i=t_i]^{u_i}\}_{i=1,\ldots,n}\end{array}}{}$$

which is the result of applying the mirroring function $\mathscr{C}$ to the derivation $\mathscr{D}$ according to the fourth clause in the inductive definition in Figure 23, is a pcu in **AK$_0$** that contains as sets of unbound consequences exactly the sets $[s_i = t_i]^{u_i}$ for $i \in \{1, \ldots, n\}$. (For this we observe that $\mathscr{C}(\mathscr{D})$ is a pcu because it has been formed from the pcu $\mathscr{C}(\mathscr{D}_1)$ by a generation step (iv) in Definition 7.1). Hence we have succeeded in performing the induction step in the case considered here.

(iii) This part of the lemma follows from the statements:

1 $\mathscr{D} \circ \mathscr{C}\,(\mathscr{D}) = \mathscr{D}$, for all derivations $\mathscr{D}$ in **e-BH$_0$**.
2 $\mathscr{C} \circ \mathscr{D}\,(\mathscr{C}) = \mathscr{C}$, for all partial consistency unfoldings $\mathscr{C}$ in **AK$_0$**.

Both of these statements can be shown by straightforward induction on $|\mathscr{D}|$ and $|\mathscr{C}|$, distinguishing as separate cases between the six inductive clauses in the definitions of $\mathscr{D}$ and $\mathscr{C}$, respectively. □

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

The 'duality theorem' below is a special case of Lemma 7.8 that reveals the relationship between $\mathbf{AK_0}$ and $\mathbf{e\text{-}BH_0}$ in the form of a direct correspondence via the mirroring functions between consistency unfoldings in $\mathbf{AK_0}$ and 'theorem-demonstrating' derivations in $\mathbf{e\text{-}BH_0}$.

Theorem 7.9. The mirroring functions $\mathscr{D}$ and $\mathscr{C}$ define a bijective functional relationship between derivations in $\mathbf{e\text{-}BH_0}$ without open assumptions and consistency unfoldings in $\mathbf{AK_0}$. More precisely, the following three assertions hold, for all $s, t \in T_\mu(\Sigma)$:

(i) For every consistency unfolding $\mathscr{C}$ of $s = t$ in $\mathbf{AK_0}$ the mirroring $\mathscr{D}(\mathscr{C})$ of $\mathscr{C}$ is a derivation in $\mathbf{e\text{-}BH_0}$ with conclusion $s = t$ and no open assumptions.

(ii) For every derivation $\mathscr{D}$ in $\mathbf{e\text{-}BH_0}$ with conclusion $s = t$ and without open assumptions, the mirroring $\mathscr{C}(\mathscr{D})$ of $\mathscr{D}$ is a consistency unfolding in $\mathbf{AK_0}$ of $s = t$.

(iii) The restrictions of the mirroring functions $\mathscr{D}$ and $\mathscr{C}$ to the set of consistency unfoldings in $\mathbf{AK_0}$ and to the set of derivations in $\mathbf{e\text{-}BH_0}$ without open assumptions, respectively, are inverses of each other.

Proof. In view of the definition of a cu in $\mathbf{AK_0}$ as a pcu without unbound consequences, statements (i) and (ii) follow as special cases of Lemma 7.8 (i) and (ii), respectively. Statement (iii) follows by Lemma 7.8 (iii), using items (i) and (ii) of the theorem. □

Example 7.10. For a binary function symbol F, consider the μ-terms over $\Sigma = \{F\}$

$$s \equiv \mu\alpha.\, F(\alpha, \gamma)$$
$$t \equiv \mu\beta.\, F(F(\beta, \gamma), \gamma),$$

which it is easy to see are equivalent. An example of a pair $\langle \mathscr{C}_1, \mathscr{D}_1 \rangle$ consisting of a consistency unfolding $\mathscr{C}_1$ of $s = t$ in $\mathbf{AK_0}$ and of a derivation in $\mathbf{e\text{-}BH_0}$ with conclusion $s = t$ and no open assumptions is shown in Figure 24. It is easy to check that the mirroring $\mathscr{D}(\mathscr{C}_1)$ of the cu $\mathscr{C}_1$ is the derivation $\mathscr{D}_1$, and that, conversely, the mirroring $\mathscr{C}(\mathscr{D}_1)$ of the derivation $\mathscr{D}_1$ coincides with the cu $\mathscr{C}_1$.

7.4. *Alternative soundness proof for* $\mathbf{BH_0}$ *and* $\mathbf{e\text{-}BH_0}$

We conclude this section by applying our duality result to give an alternative soundness proof with respect to tree unwinding equivalence for $\mathbf{e\text{-}BH_0}$, and consequently also for $\mathbf{BH_0}$. In this proof we apply the duality theorem between $\mathbf{AK_0}$ and $\mathbf{e\text{-}BH_0}$ to base the soundness of $\mathbf{e\text{-}BH_0}$ and $\mathbf{BH_0}$ on the correspondence theorem for $\mathbf{AK_0}$.

The proof below is different from both the proof of Theorem 5.3 and from a proof using 'levelled stratifications' analogous to the soundness proof by Brandt and Henglein for the axiomatisation of the subtype relation on recursive types.

Proof. (*Alternative soundness proof for* $\mathbf{BH_0}$ *and* $\mathbf{e\text{-}BH_0}$ *with respect to* $=_{\boldsymbol{T}}$.) We know from Theorem 5.9 that $\mathbf{BH_0}$ and $\mathbf{e\text{-}BH_0}$ have the same theorems, so it suffices to show the soundness with respect to $=_{\boldsymbol{T}}$ of $\mathbf{e\text{-}BH_0}$, the larger of these two systems.

Suppose that $s = t$ is a theorem of $\mathbf{e\text{-}BH_0}$. Let $\mathscr{D}$ be a derivation in $\mathbf{e\text{-}BH_0}$ with conclusion $s = t$ and no open assumptions. Then, by Theorem 7.9, the mirroring $\mathscr{C}(\mathscr{D})$ of

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

$$\mathcal{D}_1 \left\{ \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(s=t)^u \qquad \dfrac{(\text{REFL})}{\gamma=\gamma}}{F(s,\gamma)=F(t,\gamma)}\,\text{COMP}}{s=F(t,\gamma)}\,\text{FOLD}_l \qquad \dfrac{(\text{REFL})}{\gamma=\gamma}}{F(s,\gamma)=F(F(t,\gamma),\gamma)}\,\text{COMP}}{F(s,\gamma)=t}\,\text{FOLD}_r}{\mu\alpha.\,F(\alpha,\gamma)=\mu\beta.\,F(F(\beta,\gamma),\gamma)}\,\text{c-FOLD}_l,\ u \right.$$

$$\mathcal{C}_1 \left\{ \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\mu\alpha.\,F(\alpha,\gamma)=\mu\beta.\,F(F(\beta,\gamma),\gamma))^u}{F(s,\gamma)=t}\,\text{UNFOLD}_l}{F(s,\gamma)=F(F(t,\gamma),\gamma)}\,\text{UNFOLD}_r \qquad \dfrac{}{\gamma=\gamma}\,\text{DECOMP}}{s=F(t,\gamma)}\,\text{UNFOLD}_l}{F(s,\gamma)=F(t,\gamma)}\,\text{DECOMP}}{(s=t)^u \qquad \gamma=\gamma} \right.$$

Figure 24. Example consisting of a consistency unfolding $\mathscr{C}_1$ in $\mathbf{AK_0}$ and a derivation $\mathscr{D}_1$ in $\mathbf{e\text{-}BH_0}$ without open assumptions for which $\mathscr{D}(\mathscr{C}_1)=\mathscr{D}_1$ and $\mathscr{C}(\mathscr{D}_1)=\mathscr{C}_1$ where $s \equiv \mu\alpha.\,F(\alpha,\gamma)$ and $t \equiv \mu\beta.\,F(F(\beta,\gamma),\gamma)$.

$\mathscr{D}$ is a consistency unfolding of $s=t$ in $\mathbf{AK_0}$. Hence, by Lemma 7.4, the equation $s=t$ is consistent with respect to $\mathbf{AK_0}$, so Theorem 5.6 implies that s and t are equivalent. □

Although the reasoning employed in this proof can also be carried out in the opposite direction to establish completeness of $\mathbf{e\text{-}BH_0}$ with respect to $=_{\boldsymbol{T}}$ (and of $\mathbf{BH_0}$ through Theorem 5.9), this would not be conceptually different from the completeness proof for $\mathbf{BH_0}$ described earlier. This is because the proof we sketched for the '$\Leftarrow$' implication in Theorem 7.5 (which would be used in such an argument) closely matches the mirror image under the mirroring function $\mathscr{C}$ of the completeness proof for $\mathbf{BH_0}$.

7.5. *The duality result linking* $\mathbf{BH_0^{\leftrightarrows}}$ *and* $\mathbf{AK_0^{\leftrightarrows}}$

Now it is very straightforward to define, analogously to Definitions 7.1 and 7.2, the concepts of 'partial consistency unfolding' and 'consistency unfolding' in $\mathbf{AK_0^{\leftrightarrows}}$. (A motivating example for the 'consistency unfolding in $\mathbf{AK_0^{\leftrightarrows}}$' concept, which is actually less complicated than the 'consistency unfolding in $\mathbf{AK_0}$' concept, is the successful consistency check at the bottom of Figure 20.) Furthermore, we can define mirroring functions $\mathscr{D}$ and $\mathscr{C}$ between pcu's in $\mathbf{AK_0^{\leftrightarrows}}$ and derivations in $\mathbf{BH_0^{\leftrightarrows}}$ in a way that is very similar to (and in fact easier than in) Definition 7.7. In this way we are led to a counterpart of Theorem 7.9 for $\mathbf{BH_0^{\leftrightarrows}}$ and $\mathbf{AK_0^{\leftrightarrows}}$.

Theorem 7.11. There is a bijective functional relationship between derivations in $\mathbf{BH_0^{\leftrightarrows}}$ without open assumptions and consistency unfoldings in $\mathbf{AK_0^{\leftrightarrows}}$ via mirroring functions $\mathscr{C}$ and $\mathscr{D}$: this means that completely analogous statements to those in Theorem 7.9 (i), (ii) and (iii) are true.

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

$$\frac{\begin{array}{c}\mathcal{D}_1\\ s=t\end{array}}{C[s]=C[t]}\ \text{CTXT} \qquad \frac{\begin{array}{c}\big[\,C[s]=C[t]\,\big]^{u}\\ \mathcal{D}_1\\ s=t\end{array}}{C[s]=C[t]}\ \text{c-CTXT},\ u\ \ (\text{if } C\neq[\,]) \qquad \frac{C[s]=C[t]}{s=t}\ \text{CTXT}^{-1}$$

Figure 25. The context-manipulating rules CTXT, c-CTXT and CTXT^{-1} (the contexts C are assumed to be μ-free).

Now it is easy to see that Figure 20 provides an example for the assertion of this theorem concerning the ctgs's g and h of Example 2.17: it shows a pair $\langle \mathscr{D}, \mathscr{C} \rangle$ consisting of a derivation $\mathscr{D}$ in $\mathbf{BH}_{\mathbf{0}}^{\leftrightarrows}$ with conclusion $g = h$ and no open assumptions and a consistency unfolding of $g = h$ in $\mathbf{AK}_{\mathbf{0}}^{\leftrightarrows}$. Each of $\mathscr{D}$ and $\mathscr{C}$ is the mirroring of the other via mirroring functions $\mathscr{C}$ and $\mathscr{D}$, that is $\mathscr{C}(\mathscr{D}) = \mathscr{C}$ and $\mathscr{D}(\mathscr{C}) = \mathscr{D}$.

8. Concluding remarks and questions

In this concluding section we report on extensions of the duality results (without giving proofs) and pose some questions for further investigation.

A first generalisation concerns the extension of the Brandt–Henglein system **e-BH$_0$** with a context rule CTXT and a circular context rule c-CTXT, and the extension of the Ariola–Klop system **AK$_0$** with a dual rule CTXT^{-1} (see Figure 25). All occurring contexts are assumed to be *μ-free* here, that is, they do not contain μ-bindings. Applications of the circular inference rule c-CTXT are subject to the side-condition that the context C is not the trivial context $[\,]$.

Using a proof similar to that of Theorem 5.9, we can show that the extension of **e-BH$_0$** with the rules CTXT and c-CTXT is sound for tree unwinding equivalence. Furthermore, it is straightforward to extend the concept consistency unfolding to the system **AK$_0$**+CTXT^{-1} resulting from **AK$_0$** by the addition of the CTXT^{-1} rule , and to extend the mirroring functions $\mathscr{C}$ and $\mathscr{D}$ of Definition 7.7 to functions between **AK$_0$**+CTXT^{-1} and the extension **e-BH$_0$**+(c-)CTXT of **e-BH$_0$**. With these preparations, we can prove the following extension of Theorem 7.9.

Theorem 8.1. There is a duality between derivations in **e-BH$_0$**+(c-)CTXT without open assumptions and consistency unfoldings in **AK$_0$**+CTXT^{-1} via extensions of the mirroring functions $\mathscr{C}$ and $\mathscr{D}$.

The example given in Figure 26 illustrates this duality in the form of a pair consisting of a consistency unfolding $\mathscr{C}$ in **AK$_0$**+CTXT^{-1} and a derivation without open assumptions in **e-BH$_0$**+CTXT such that $\mathscr{C}$ and $\mathscr{D}$ are mirror images of each other under extensions of the mirroring functions $\mathscr{D}$ and $\mathscr{C}$. Note that $\mathscr{C}$ and $\mathscr{D}$ here are smaller than the cu $\mathscr{C}_1$ in **AK$_0$** and the **e-BH$_0$**-derivation $\mathscr{D}_1$ in Figure 24, which establish corresponding provability and consistency statements for the same μ-terms.

A second extension of our duality result concerns systems that arise by adding inference rules that describe substitutions into other μ-terms, generalising the context rules for μ-free contexts considered above. The SUB-INTO rule, its circular version c-SUB-INTO and

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(s=t)^u}{F(s,\gamma)=t}\,\mathrm{UNFOLD}_l}{F(s,\gamma)=F(F(t,\gamma),\gamma)}\,\mathrm{UNFOLD}_r}{s=F(t,\gamma)}\,\mathrm{CTXT}^{-1}}{F(s,\gamma)=F(t,\gamma)}\,\mathrm{UNFOLD}_l}{(s=t)^u}\,\mathrm{CTXT}^{-1} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(s=t)^u}{F(s,\gamma)=F(t,\gamma)}\,\mathrm{CTXT}}{s=F(t,\gamma)}\,\mathrm{FOLD}_l}{F(s,\gamma)=F(F(t,\gamma),\gamma)}\,\mathrm{CTXT}}{F(s,\gamma)=t}\,\mathrm{FOLD}_r}{(s=t)^u}\,\text{c-FOLD}_l,\,u$$

Figure 26. A consistency unfolding $\mathscr{C}$ in $\mathbf{AK_0}$+CTXT^{-1} (on the left) and a derivation $\mathscr{D}$ in $\mathbf{e\text{-}BH_0}$+CTXT (on the right) that are mirror images of each other. Here $s \equiv \mu\alpha.\,F(\alpha,\gamma)$ and $t \equiv \mu\beta.\,F(F(\alpha,\gamma),\gamma)$ are the μ-terms used in Example 7.10 and Figure 24.

$$\dfrac{\dfrac{\mathcal{D}_1}{s=t}}{r[s/\alpha]=r[t/\alpha]}\,\mathrm{SUB\text{-}INTO} \qquad \dfrac{\dfrac{\big[\,r[s/\alpha]=r[t/\alpha]\,\big]^u}{\mathcal{D}_1}}{r[s/\alpha]=r[t/\alpha]}\,\text{c-SUB-INTO},\,u\ (\mathcal{L}(r)\neq\alpha,\ \alpha\text{ free in }r)$$

$$\dfrac{r[s/\alpha]=r[t/\alpha]}{s=t}\,(\mathrm{SUB\text{-}INTO})^{-1}$$

Figure 27. The SUB-INTO rule for substitution into a μ-term, the dual (SUB-INTO)$^{-1}$ rule and c-SUB-INTO, the circular version of SUB-INTO.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{s=t}{(F(s,\gamma)=F(F(t,\gamma),\gamma))^u}\,\mathrm{UNFOLD}_{l/r}}{s=F(t,\gamma)}\,(\mathrm{SUB\text{-}INTO})^{-1}}{F(s,\gamma)=F(t,\gamma)}\,\mathrm{UNFOLD}_l}{s=t}\,(\mathrm{SUB\text{-}INTO})^{-1}}{(F(s,\gamma)=F(F(t,\gamma),\gamma))^u}\,\mathrm{UNFOLD}_{l/r} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(F(s,\gamma)=F(F(t,\gamma),\gamma))^u}{s=t}\,\mathrm{FOLD}_{l/r}}{F(s,\gamma)=F(t,\gamma)}\,\mathrm{SUB\text{-}INTO}}{s=F(t,\gamma)}\,\mathrm{FOLD}_l}{F(s,\gamma)=F(F(t,\gamma),\gamma)}\,\text{c-SUB-INTO},\,u}{s=t}\,\mathrm{FOLD}_{l/r}$$

Figure 28. A consistency unfolding $\mathscr{C}$ in $\mathbf{AK_0}$+(SUB-INTO)$^{-1}$ (on the left) and a derivation $\mathscr{D}$ in $\mathbf{e\text{-}BH_0}$+(c-)SUB-INTO (on the right) that are mirror images of each other. Here $s \equiv \mu\alpha.\,F(\alpha,\gamma)$ and $t \equiv \mu\beta.\,F(F(\alpha,\gamma),\gamma)$ from Example 7.10 and Figure 24 are used.

the dual rule (SUB-INTO)$^{-1}$ shown in Figure 27 generalise the context rules considered above (which only account for substitution into μ-free μ-terms). The following duality result can be shown by:

(1) showing the soundness of $\mathbf{e\text{-}BH_0}$+(c-)SUB-INTO;
(2) introducing consistency unfoldings also for the system $\mathbf{AK_0}$+(SUB-INTO)$^{-1}$; and
(3) extending the mirroring functions $\mathscr{C}$ and $\mathscr{D}$ appropriately.

Theorem 8.2. There is a duality between derivations in $\mathbf{e\text{-}BH_0}$+(c-)SUB-INTO without open assumptions and consistency unfoldings in $\mathbf{AK_0}$+(SUB-INTO)$^{-1}$ via extensions of the mirroring functions $\mathscr{C}$ and $\mathscr{D}$.

Figure 28 shows a cu $\mathscr{C}$ in $\mathbf{AK_0}$+(SUB-INTO)$^{-1}$ and a derivation without open assumptions in $\mathbf{e\text{-}BH_0}$+(c-)SUB-INTO that are mirror images of each other.

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

$$\frac{\mathcal{D}_1}{\dfrac{s = t}{\sigma(s) = \sigma(t)}}\ \text{SUBST} \qquad \frac{\mathcal{D}_1}{\dfrac{s = t}{s[r/\alpha] = t[r/\alpha]}}\ \text{SUB} \qquad \frac{s[r/\alpha] = t[r/\alpha]}{s = t}\ (\text{SUB})^{-1}$$

Figure 29. The substitution rule SUBST, a simplified version SUB and its dual rule $(\text{SUB})^{-1}$ (where σ denotes a substitution on μ-terms).

A similar duality result is possible for an extension of $\mathbf{e\text{-}BH_0}$ with each of the substitution rules SUBST and SUB in relation to an extension of $\mathbf{AK_0}$ with the dual rule $(\text{SUB})^{-1}$ (see Figure 29 for these rules and note that SUB can be used to mimic SUBST). It is interesting that consistency unfoldings in $\mathbf{AK_0}+(\text{SUB})^{-1}$ (which can be defined in a straighforward manner) seem to correspond to reasoning that is frequently employed when using Goguen and Rosu's 'circular coinduction' method (Rosu and Goguen 2001).

The following may be a related issue. We have observed that derivations in $\mathbf{BH_0}$ without open assumptions and consistency unfoldings in $\mathbf{AK_0}$ formalise bisimulations on μ-terms (up to adding $\to_{\text{ou}}$-reachable pairs). Furthermore, derivations in the Brandt–Henglein system **BH**, which contains symmetry and transitivity rules, correspond to bisimulations up to symmetry and transitivity. The concept of 'bisimulation up to' was first introduced in process theory, where 'up-to results' turned out to be useful in reducing the work required to show that specific processes are bisimilar. Sangiorgi (1998) provides an illuminating abstract treatment of 'up-to techniques'.

For the additional rules considered in this section, derivations without open assumptions in $\mathbf{BH_0}+$(c-)SUB-INTO and $\mathbf{BH_0}+$SUB correspond to bisimulations on μ-terms up to substitution into μ-terms and up to substitution, respectively. We believe that it would be worthwhile studying the connection with up-to techniques in greater depth.

Question 8.3. *What kind of abstract duality results can be obtained by relating general up-to techniques to rules in coinductive proof systems for proving bisimilarity, on the one hand, and to dual rules in corresponding syntactic-matching systems, on the other?*

The connection with the 'Recursion Inference Rule' RIR in Moschovakis's FLR_0 ('formal language of recursion') is also interesting – see, for instance, the treatments in Hurkens *et al.* (1998) and Moss (2003). Formulas in FLR_0 are nested equational specifications, of which ctgs's are only a very special case. There seems to be a conceptual difference between FLR_0 and Brandt–Henglein systems, due to the fact that single applications of the RIR rule in FLR_0 may amount to the choice of a bisimulation as a whole, while, for instance, in $\mathbf{BH_0}$ only entire derivations without open assumptions correspond to bisimulations. This fact deserves some further investigation on the proof-theoretic level.

The proof systems studied here have been concerned exclusively with showing that two cyclic objects are equivalent in a definite sense. It would also be interesting to investigate proof systems concerned with notions of equivalence between functions on cyclic objects.

Question 8.4. *Do the duality results presented here generalise, in some sense, to proof systems concerned with notions of equivalence between functions (and perhaps higher-order*

https://doi.org/10.1017/S0960129507006111 Published online by Cambridge University Press

functions) on cyclic objects? Are there duality results for proof systems concerned with proving equality in final coalgebras? For example, for proving equality of streams and of stream functions (built from well-known stream functions such as even, odd *and* zip*)?*

Acknowledgments

I would like to thank Jan Willem Klop for suggesting an investigation into the proof-theoretic connections between proof systems for cyclic objects, for a basic idea and for helping with the restructuring of this article. I am also indebted to Bas Luttik for carefully reading drafts of Grabmayer (2002a), and for offering many suggestions.

Finally, I would like to thank the anonymous referees for their careful reading, detailed observations and suggestions concerning both immediate improvements and ideas for the future.

References

Amadio, R. M. and Cardelli, L. (1993) Subtyping recursive types. *ACM Transactions on Programming Languages and Systems* **15** (4) 575–631.

Ariola, Z. M. and Klop, J. W. (1995) Equational term graph rewriting. Technical Report IR-391, Vrije Universiteit Amsterdam. (This is an extension of Ariola and Klop (1996).)

Ariola, Z. M. and Klop, J. W. (1996) Equational term graph rewriting. *Fundamenta Informaticae* **26** (3-4) 207–240.

Blom, S. (2001) *Term Graph Rewriting, Syntax and Sematics*, Ph.D. thesis, Vrije Universiteit Amsterdam.

Brandt, M. and Henglein, F. (1998) Coinductive axiomatization of recursive type equality and subtyping. *Fundamenta Informaticae* **33** 1–30.

Grabmayer, C. (2002a) A duality in proof systems for recursive type equality and bisimulation equivalence on cyclic term graphs. In: Plump, D. (ed.) Proceedings of TERMGRAPH 2002. *Electronic Notes in Computer Science* **72** (1).

Grabmayer, C. (2002b) A duality in proof systems for recursive type equality and bisimulation equivalence on cyclic term graphs. Technical Report IR-499, Dept. of Math. and Comp. Sci., Vrije Universiteit Amsterdam. (Available from the authors's home page.)

Grabmayer, C. (2005) *Relating Proof Systems for Recursive Types*, Ph.D. thesis, Vrije Universiteit Amsterdam.

Hurkens, A. J. C., McArthur, M., Moschovakis, Y., Moss, L. S. and Whitney, G. (1998) The logic of recursive equations. *Journal of Symbolic Logic* **63** (2) 451–478.

Hüttel, H. and Stirling, C. (1991) Actions speak louder than words: Proving bisimilarity for context-free processes. In: *Proceedings of LICS'91*, IEEE Computer Society Press 376–386.

Klop, J. W. (2000) Proof systems for cyclic term graphs. Lecture at the Winter Workshop on Logics, Types and Rewriting, February 1-3 2000, Heriot-Watt University, Edinburgh. (Available at: `http://www.cs.vu.nl/~jwk/ctg1-41.pdf`.)

Milner, R. (1984) A complete inference system for a class of regular behaviours. *Journal of Computer and System Sciences* **28** 439–466.

Moss, L. S. (2003) Recursion and corecursion have the same equational logic. *Theoretical Computer Science* **294** (1-2) 233–267.

Rosu, G. and Goguen, J. (2001) Circular coinduction. In: *Proceedings of IJCAR'01, Siena, Italy*.

Salomaa, A. (1966) Two complete axiom systems for the algebra of regular events. *Journal of the ACM* **13** (13) 158–169.

Sangiorgi, D. (1998) On the bisimulation proof method. *Mathematical Structures in Computer Science* **8** (5) 447–479.

Troelstra, A. and Schwichtenberg, H. (2000) *Basic Proof Theory*, Cambridge University Press.