

Diacritical Companions

Dariusz Biernacki^{a,1} Sergueï Lenglet^{b,2} Piotr Polesiuk^{a,3}

^a *Institute of Computer Science, University of Wrocław, Wrocław, Poland*

^b *Loria, Université de Lorraine, Nancy, France*

Abstract

Coinductive reasoning in terms of bisimulations is in practice routinely supported by carefully crafted up-to techniques that can greatly simplify proofs. However, designing and proving such bisimulation enhancements sound can be challenging, especially when striving for modularity. In this article, we present a theory of up-to techniques that builds on the notion of companion introduced by Pous and that extends our previous work which allows for powerful up-to techniques defined in terms of diacritical progress of relations. The theory of diacritical companion that we put forward works in any complete lattice and makes it possible to modularly prove soundness of up-to techniques which rely on the distinction between passive and active progresses, such as up to context in λ -calculi with control operators and extensionality.

Keywords: coinduction, bisimulation, up-to technique, diacritical progress, companion

1 Introduction

Coinduction allows to define and reason about potentially infinite objects, such as streams [14] or program behaviors [8]. Proving a property by coinduction consists in exhibiting an *invariant*, known as *bisimulation* in the context of program equivalence. Finding the appropriate invariant for a given property can be challenging, as it has to be large enough for the invariant to hold, but also small enough to keep the proof as simple as possible. *Up-to techniques* or *enhancements* [15,13] have been introduced to relieve this tension, as they permit to consider smaller objects which are not invariant but contained into one. For example, in some languages, it is safe to ignore a common context when proving the equivalence of two given programs [4]. This reasoning *up to context* is not sound in general, so the problem becomes how to prove that a given technique is indeed valid, especially since the composition of two up-to techniques is not necessarily sound.

¹ Email: dabi@cs.uni.wroc.pl

² Email: serguei.lenglet@univ-lorraine.fr

³ Email: ppolesiuk@cs.uni.wroc.pl

Sangiorgi and Pous studied up-to techniques first for binary relations [15], and then in the more general setting of complete lattices [13], focusing on finding composable enhancements for better modularity. Their work led to the definition of the subclass of *compatible* enhancements, which is closed by function composition and union. Following up on a remark by Hur et al. [6], Pous [11] shows how useful the greatest compatible function (the *companion*) can be used as a sufficient, modular criterion for the validity of up-to techniques. Indeed, any function (compatible or not) below the companion is an up-to technique, and being below the companion is preserved by composition and union. As further demonstrated by Pous, the notion of companion is surrounded by an array of tools that in practice are highly flexible and effective, compared to relying on the notion of compatibility alone. The companion has since been studied as a categorical object [12,2] and has been characterized in a classical [9] and constructive [16] setting. Danielsson [5] uses the former characterization to relate the companion to size-preserving functions, a class of up-to techniques proved valid using types.

In our studies of equivalences for variants of the λ -calculus [1,3], we restrict the use of up-to techniques in some cases, to allow for more powerful enhancements which would otherwise be unsound when applied in an unconstrained way. We then distinguish the *strong* up-to techniques, which can safely be used with no restrictions, from the regular ones with a restricted use. This distinction makes it possible to define a highly expressive variant of environmental bisimulation up to context, particularly useful in the presence of context-manipulating constructs such as control operators [1]. It also enables up-to techniques for normal-form bisimulation which respect η -expansion [3]. In the aforementioned papers, we propose *diacritical compatibility* as a sufficient criterion to be a (strong) up-to technique in our setting. This notion does not allow for an interesting notion of companion, as it is defined at the level of sets of functions and not for individual functions.

This paper fixes this issue by revisiting our view of up-to techniques: instead of distinguishing between strong and regular techniques, we decompose an up-to technique into a pair, composed of a safe (or *strong*) part, which can be used everywhere, and a restricted (or *weak*) one, which cannot. We then define compatibility for pairs of functions; the corresponding companion is therefore itself a pair, called the *diacritical companion*. We show that the diacritical companion enjoys the same properties as Pous' companion, and can be used as a sufficient criterion for the validity of up-to techniques. In contrast with our previous papers, the current developments are not restricted to binary relations and take place in any complete lattice. The theory of this article therefore extends our previous work as well as Pous'.

In the rest of the paper, we first introduce the concepts and notions used throughout this work in Section 2. We define the diacritical companion in Section 3, and then extend it to the higher-order setting to use it as a proof technique to show that a given pair of functions is below the first-order diacritical companion (Section 4). We conclude in Section 5. The theory (not the examples) has been formalized in the Coq proof assistant; the developments are available at <https://github.com/danielsanku/diacritical-companion>.

[//bitbucket.org/pl-uwcr/companion/](http://bitbucket.org/pl-uwcr/companion/).

2 Diacritical Progress, Complete Lattices, and Companion

We set the stage by reminding the notion of diacritical progress, usual definitions about complete lattices, and Pous' companion.

2.1 Diacritical Progress

The notion of *progress* [15] between relations makes it possible to uniformly describe bisimulations and up-to techniques. A relation \mathcal{R} progresses to \mathcal{S} , usually written $\mathcal{R} \mapsto \mathcal{S}$, if terms related by \mathcal{R} can be turned into terms related by \mathcal{S} by following some conditions given by \mapsto . A *simulation* is then a relation which progresses to itself, i.e., $\mathcal{R} \mapsto \mathcal{R}$; a *bisimulation* \mathcal{R} is a relation such that \mathcal{R} and its inverse \mathcal{R}^{-1} are simulations, and *bisimilarity* is the largest bisimulation. An *up-to technique* or *enhancement* is a function f on relations such that $\mathcal{R} \mapsto f(\mathcal{R})$ implies that \mathcal{R} is included in the bisimilarity.

Example 2.1 All along the paper, we use normal-form bisimilarity [7] for the λ -calculus as an illustrative example. We let x, y, z range over variables. The syntax of terms (e), values (v), and evaluation contexts (E) is given by:

$$e ::= v \mid e e \quad v ::= x \mid \lambda x.e \quad E ::= \square \mid E e \mid v E$$

An abstraction $\lambda x.e$ binds x in e ; a variable that is not bound is called free. We work modulo α -conversion of bound variables, and a variable is called fresh if it does not occur in the terms under consideration. We write $E[e]$ for plugging a term in an evaluation context, and $e\{v/x\}$ for the capture-avoiding substitution of v for x in e , defined as usual. We abbreviate successive λ -abstractions $\lambda x.\lambda y.e$ as $\lambda xy.e$. The call-by-value reduction semantics is given by the following rule.

$$E[(\lambda x.e) v] \rightarrow E[e\{v/x\}]$$

We write \rightarrow^* for the reflexive and transitive closure of \rightarrow .

Normal-form bisimilarity [7] relates terms by reducing them to normal forms, which are then decomposed into bisimilar subterms. With the chosen semantics, normal forms are either values or *open-stuck terms* of the form $E[x v]$. The notion of progress corresponding to normal-form bisimilarity can be given as follows: $\mathcal{R} \mapsto \mathcal{S}$ if $\mathcal{R} \subseteq \mathcal{S}$ and $e_1 \mathcal{R} e_2$ implies:

- (i) if $e_1 \rightarrow e'_1$ then there exists e'_2 such that $e_2 \rightarrow^* e'_2$ and $e'_1 \mathcal{S} e'_2$;
- (ii) if $e_1 = v_1$ then there exists v_2 such that $e_2 \rightarrow^* v_2$ and for a fresh variable z , we have $v_1 z \mathcal{S} v_2 z$;
- (iii) if $e_1 = E_1[x v_1]$ then there exist E_2 and v_2 such that $e_2 \rightarrow^* E_2[x v_2]$ and for a fresh variable z , we have $E_1[z] \mathcal{S} E_2[z]$ and $v_1 z \mathcal{S} v_2 z$.

Clause (ii) allows for a uniform treatment of values by simply applying them to a fresh variable. Such a formulation respects η -expansion and scales to richer calculi with, e.g., control operators [3]. \square

In our previous work [1,3], we decomposed progress into a *passive* and an *active* part, with the idea that passive progress covers administrative steps, and active progress actual computation steps. *Diacritical progress* $\mathcal{R} \rightsquigarrow \mathcal{Q}$, \mathcal{S} means that \mathcal{R} progresses towards \mathcal{Q} in the passive case, and towards \mathcal{S} in the active one. The resulting notions of simulation and bisimilarity are not affected by this change, since a simulation still progresses to itself in the active and passive cases: $\mathcal{R} \rightsquigarrow \mathcal{R}$, \mathcal{R} .

Instead, diacritical progress allows for a finer control on enhancements by forbidding the application of some of them in the passive case. We distinguish between regular techniques, which can be used only in the active case, from the strong ones usable everywhere. Formally, a function f is an up-to technique if $\mathcal{R} \rightsquigarrow \mathcal{R}$, $f(\mathcal{R})$ implies that \mathcal{R} is included in the bisimilarity, and it is a strong one if $\mathcal{R} \rightsquigarrow f(\mathcal{R})$, $f(\mathcal{R})$ implies the same property. Such a distinction enables techniques that otherwise would be unsound if applied in an unconstrained way.

Example 2.2 For normal-form bisimilarity [3], we consider the clauses (i) and (iii) as active, since computation happens in both cases, either explicitly in (i), or implicitly in (iii). Indeed, clause (iii) runs the surrounding context E with a variable z which plays the role of a value obtained by at least one reduction step of $x v_1$, if x was replaced with a λ -abstraction. In contrast, value testing (ii) is passive, as it simply builds terms and does not reduce them.

W.r.t. up-to techniques, such a choice allows to forbid in particular the factorization of a common context when comparing values, as authorizing such a technique in that case would be unsound: $v_1 \mathcal{R} v_2$ implies that $v_1 z$ and $v_2 z$ are related up to context for any z , v_1 , and v_2 . Up to context can still be safely used when comparing open-stuck terms or after a reduction step. \square

In our previous work, we proposed a sufficient criterion to show that a function is a (strong) up-to technique, phrased in terms of evolution and compatibility. A function f *diacritically evolves* to g and h if $\mathcal{R} \rightsquigarrow \mathcal{R}$, \mathcal{S} implies $f(\mathcal{R}) \rightsquigarrow g(\mathcal{R})$, $h(\mathcal{S})$, and it *strongly evolves* to g and h if $\mathcal{R} \rightsquigarrow \mathcal{Q}$, \mathcal{S} implies $f(\mathcal{R}) \rightsquigarrow g(\mathcal{Q})$, $h(\mathcal{S})$. These notions mean that f is transformed into g in the passive case, and into h in the active one. Diacritical evolution makes the extra hypothesis that \mathcal{R} passively progresses to itself, so that we can normalize the elements in \mathcal{R} w.r.t. passive progress before making an active progress. Compatibility is then defined for sets of functions: roughly, a set \mathbb{F} is compatible if it contains a subset \mathbb{S} such that

- every function f of \mathbb{S} strongly evolves towards g and h such that g is built from \mathbb{S} (using composition and union), and h is built from \mathbb{F} ;
- every function f of \mathbb{F} diacritically evolves towards $g_1 \circ f \circ g_2$ and h such that g_1 and g_2 are built from \mathbb{S} and h is built from \mathbb{F} .

If \mathbb{F} is compatible, then every function in \mathbb{F} is an up-to technique, and every function in \mathbb{S} is a strong up-to technique. In both cases, h can be built out of functions in \mathbb{F} ,

meaning that any up-to technique can be used in the active case, while only strong techniques can be used in the passive case to build \mathbf{g} , \mathbf{g}_1 , or \mathbf{g}_2 .

Such a definition of compatibility does not lend itself to a notion of companion, because it is defined at the level of sets and not individual functions. We propose a much cleaner theory in this paper, by defining a notion of compatibility and its corresponding companion which allows for the distinction between active and passive progress while extending Pous' framework.

2.2 Progress for Complete Lattices

The notion of progress underlying a bisimilarity can also be expressed as a monotone function \mathbf{b} on relations [13]; bisimulations are then *post-fixpoints* of \mathbf{b} , i.e., relations \mathcal{R} verifying $\mathcal{R} \subseteq \mathbf{b}(\mathcal{R})$. Bisimilarity is the greatest fixpoint of \mathbf{b} , and can be characterized by Knaster-Tarski's theorem as the union of all bisimulations. Pous defines the companion for any complete lattice for a given monotone function [11].

However, the notion of progress can also be generalized to any complete lattice [10], with a one-to-one correspondence to monotone functions, meaning that we still benefit from Knaster-Tarski's theorem. We present our work using this notion of progress, as we believe it simplifies the presentation of the theory and its comprehension. In particular, our examples relying on normal-form bisimilarity can still be formulated in terms of progress and do not need a less natural translation to monotone functions. To be consistent with this choice, we use progress-like notations and terminology all along the paper, like, e.g., simulation for post-fixpoint and similarity for greatest fixpoint.

We remind some general notions and fix some notations. A *complete lattice* consists of a set \mathbb{L} (whose elements are ranged over by $\mathcal{R}, \mathcal{S}, \mathcal{Q}$) equipped with a partial order \sqsubseteq such that any subset \mathbb{X} of \mathbb{L} has a least upper bound $\bigsqcup \mathbb{X}$ verifying: for all $\mathcal{R} \in \mathbb{L}$, $\bigsqcup \mathbb{X} \sqsubseteq \mathcal{R}$ iff for all $\mathcal{S} \in \mathbb{X}$, $\mathcal{S} \sqsubseteq \mathcal{R}$. The least upper bound of two elements \mathcal{R} and \mathcal{S} is written $\mathcal{R} \sqcup \mathcal{S}$, and we write \perp for the bottom element of \mathbb{L} , defined as $\bigsqcup \emptyset$. Given a predicate \mathcal{P} on elements of \mathbb{L} , we write $\bigsqcup \mathcal{R}.\mathcal{P}(\mathcal{R})$ for $\bigsqcup \{\mathcal{R} \mid \mathcal{P}(\mathcal{R})\}$. By inverting the ordering, we derive the definition of the greatest lower bound \sqcap of a set, including the binary one \sqcap , as well as the top element \top .

A function \mathbf{f} from \mathbb{L} to \mathbb{L} is *monotone* if for all x, y , $x \sqsubseteq y$ implies $\mathbf{f}(x) \sqsubseteq \mathbf{f}(y)$. We write $[\mathbb{L} \rightarrow \mathbb{L}]$ for the set of monotone functions from \mathbb{L} to \mathbb{L} , and denote its elements with either $\mathbf{f}, \mathbf{g}, \mathbf{h}, \mathbf{b}$, or \mathbf{s} . The set $[\mathbb{L} \rightarrow \mathbb{L}]$ forms a complete lattice if we extend \sqsubseteq and \bigsqcup as follows: $\mathbf{f} \sqsubseteq \mathbf{g}$ if for all $x \in \mathbb{L}$, $\mathbf{f}(x) \sqsubseteq \mathbf{g}(x)$, and for all $\mathbb{X} \subseteq [\mathbb{L} \rightarrow \mathbb{L}]$, $\bigsqcup \mathbb{X} \triangleq x \mapsto \bigsqcup \mathbf{f}(x).\mathbf{f} \in \mathbb{X}$. We write id for the identity function. Given two complete lattices \mathbb{L}_1 and \mathbb{L}_2 , their product $\mathbb{L}_1 \times \mathbb{L}_2$, ordered pointwise, is also a complete lattice. We write π_1 and π_2 for respectively the first and second projection.

Given a complete lattice \mathbb{L} , the general definition of progress [10] describes how to go from one element of \mathbb{L} to another, respecting the ordering \sqsubseteq and the least upper bound \bigsqcup .

Definition 2.3 A binary relation \rhd on \mathbb{L} is a *progress relation* if the following holds:

- (i) if $\mathcal{R} \rightarrow \mathcal{Q}$ and $\mathcal{Q} \sqsubseteq \mathcal{S}$ then $\mathcal{R} \rightarrow \mathcal{S}$;
- (ii) if $\mathcal{R} \sqsubseteq \mathcal{Q}$ and $\mathcal{Q} \rightarrow \mathcal{S}$ then $\mathcal{R} \rightarrow \mathcal{S}$;
- (iii) for any subset \mathbb{X} of \mathbb{L} , if for each $\mathcal{R} \in \mathbb{X}$ we have $\mathcal{R} \rightarrow \mathcal{S}$, then $\bigsqcup \mathbb{X} \rightarrow \mathcal{S}$.

Least upper bounds are preserved by \rightarrow on the left, but also on the right, as a consequence of the first item. Indeed, given a subset \mathbb{X} of \mathbb{L} , if $\mathcal{S} \in \mathbb{X}$ implies $\mathcal{R} \rightarrow \mathcal{S}$, then since $\mathcal{S} \sqsubseteq \bigsqcup \mathbb{X}$, we also have $\mathcal{R} \rightarrow \bigsqcup \mathbb{X}$.

Progress relations correspond to monotone functions since we can define a progress $\xrightarrow{\mathbf{b}}$ from a given monotone function \mathbf{b} , and vice versa:

$$\mathcal{R} \xrightarrow{\mathbf{b}} \mathcal{S} \triangleq \mathcal{R} \sqsubseteq \mathbf{b}(\mathcal{S}) \qquad \mathbf{b}(\mathcal{S}) \triangleq \bigsqcup \mathcal{R}. \mathcal{R} \xrightarrow{\mathbf{b}} \mathcal{S}.$$

A progress relation $\xrightarrow{\mathbf{b}}$ on \mathbb{L} induces its counterpart in $[\mathbb{L} \rightarrow \mathbb{L}]$, called *\mathbf{b} -evolution*.

Definition 2.4 Let $\xrightarrow{\mathbf{b}}$ be a progress relation on \mathbb{L} , and $(\mathbf{f}, \mathbf{g}) \in [\mathbb{L} \rightarrow \mathbb{L}]^2$. The function \mathbf{f} *\mathbf{b} -evolves* to \mathbf{g} , written $\mathbf{f} \xrightarrow{\mathbf{b}} \mathbf{g}$, if for all $(\mathcal{R}, \mathcal{S}) \in \mathbb{L}^2$, $\mathcal{R} \xrightarrow{\mathbf{b}} \mathcal{S}$ implies $\mathbf{f}(\mathcal{R}) \xrightarrow{\mathbf{b}} \mathbf{g}(\mathcal{S})$.

Evolution is a progress relation on $[\mathbb{L} \rightarrow \mathbb{L}]$, and is preserved by composition.

Proposition 2.5 If $\mathbf{f}_1 \xrightarrow{\mathbf{b}} \mathbf{g}_1$ and $\mathbf{f}_2 \xrightarrow{\mathbf{b}} \mathbf{g}_2$ then $\mathbf{f}_1 \circ \mathbf{f}_2 \xrightarrow{\mathbf{b}} \mathbf{g}_1 \circ \mathbf{g}_2$.

2.3 Companion

We briefly remind the definition and the main properties of Pous' companion [11], reformulated using progress and evolution to be consistent with the rest of the paper. Let \mathbb{L} be a complete lattice, and $\xrightarrow{\mathbf{b}}$ a progress relation on \mathbb{L} , for which we consider the corresponding notion of similarity, written $\nu(\xrightarrow{\mathbf{b}})$.

A *compatible function* for $\xrightarrow{\mathbf{b}}$ is a monotone function \mathbf{f} such that $\mathbf{f} \xrightarrow{\mathbf{b}} \mathbf{f}$, and the *companion*, written \mathbf{t} , is the largest compatible function. What makes \mathbf{t} interesting is that it is an up-to technique, and any function below \mathbf{t} is an up-to technique.

Theorem 2.6 Let $\mathcal{R} \in \mathbb{L}$. If $\mathcal{R} \xrightarrow{\mathbf{b}} \mathbf{t}(\mathcal{R})$ then $\mathcal{R} \sqsubseteq \nu(\xrightarrow{\mathbf{b}})$.

Corollary 2.7 Let $\mathbf{f} \in [\mathbb{L} \rightarrow \mathbb{L}]$. If $\mathbf{f} \sqsubseteq \mathbf{t}$ then \mathbf{f} is an up-to technique.

To show that \mathbf{f} is below \mathbf{t} , we just need to consider the lattice $[\mathbb{L} \rightarrow \mathbb{L}]$ and the progress relation $\xrightarrow{\mathbf{b}}$, whose simulations and similarity are, respectively, compatible functions and \mathbf{t} . The *higher-order companion* \mathbf{T} is then the companion for $\xrightarrow{\mathbf{b}}$, and instantiating Theorem 2.6 in that setting gives the following result.

Corollary 2.8 Let $\mathbf{f} \in [\mathbb{L} \rightarrow \mathbb{L}]$. If $\mathbf{f} \xrightarrow{\mathbf{b}} \mathbf{T}(\mathbf{f})$ then $\mathbf{f} \sqsubseteq \mathbf{t}$.

The \mathbf{b} -evolution proof for a given \mathbf{f} can then be simplified thanks to the following properties of \mathbf{T} , where we write \mathbf{b} for the monotone function corresponding to $\xrightarrow{\mathbf{b}}$.

Proposition 2.9 Let $f \in [\mathbb{L} \rightarrow \mathbb{L}]$. The following holds:

$$\begin{array}{lll} \text{id} \sqsubseteq \mathbf{T}(f), & f \sqsubseteq \mathbf{T}(f), & \mathbf{T}(\mathbf{T}(f)) \sqsubseteq \mathbf{T}(f), \\ b \sqsubseteq \mathbf{T}(f), & t \sqsubseteq \mathbf{T}(f), & \mathbf{T}(f) \circ \mathbf{T}(f) \sqsubseteq \mathbf{T}(f). \end{array}$$

Our goal is to tell a similar story for diacritical progress. We start in the following section by defining a companion for which we prove a result similar to Theorem 2.6.

3 First-order Diacritical Companion

Henceforth, we fix a complete lattice \mathbb{L} with order \sqsubseteq and least upper bound \sqcup , and two progress relations \succ^p and \succ^a , referred to as the passive and active progress, respectively. Diacritical progress \succ is the conjunction of the two progresses, and a simulation for \succ is therefore a simulation for \succ^p and for \succ^a . We write $\nu(\succ^p, \succ^a)$ for the corresponding similarity.

Example 3.1 The passive and active progresses for normal-form similarity are defined as follows. Let \mathcal{R}, \mathcal{S} be relations on λ -terms.

- We write $\mathcal{R} \succ^p \mathcal{S}$ if $\mathcal{R} \subseteq \mathcal{S}$ and $v_1 \mathcal{R} e_2$ implies that there exists v_2 such that $e_2 \rightarrow^* v_2$, and $v_1 z \mathcal{S} v_2 z$ for a fresh variable z .
- We write $\mathcal{R} \succ^a \mathcal{S}$ if $\mathcal{R} \subseteq \mathcal{S}$ and $e_1 \mathcal{R} e_2$ implies
 - if $e_1 \rightarrow e'_1$, then there exists e'_2 such that $e_2 \rightarrow^* e'_2$ and $e'_1 \mathcal{S} e'_2$;
 - if $e_1 = E_1[x v_1]$, then there exist E_2, v_2 such that $e_2 \rightarrow^* E_2[x v_2]$, $E_1[z] \mathcal{S} E_2[z]$, and $v_1 z \mathcal{S} v_2 z$ for a fresh variable z .

We discuss the need for the inclusion hypothesis $\mathcal{R} \subseteq \mathcal{S}$ and its consequences in Remark 4.9. \square

In our previous work, we defined compatibility not simply for functions but for sets of functions, to allow for a strong candidate to evolve towards a regular one in the active progress case, as we can see with the definition of strong evolution in Section 2.1. The same purpose can be achieved by decomposing up-to techniques into *pairs* of monotone functions (s, f) , where s is the function applied after the passive progress, while f is applied after the active one.

Definition 3.2 Let $(s, f) \in [\mathbb{L} \rightarrow \mathbb{L}]^2$; (s, f) is an up-to technique if for all $\mathcal{R} \in \mathbb{L}$, $\mathcal{R} \succ^p s(\mathcal{R})$ and $\mathcal{R} \succ^a f(\mathcal{R})$ imply $\mathcal{R} \sqsubseteq \nu(\succ^p, \succ^a)$.

By considering pairs of functions, we generalize the distinction between strong and regular up-to techniques described in Section 2.1; we decompose an up-to technique into its *strong* and *weak* components, respectively s and f . Such a change allows for a definition of evolution between two pairs (s, f) and (s', f') where s evolves towards f' in the active progress case.

The function f can be seen as weak since it requires an extra hypothesis in its evolution proof, i.e., it requires arguments that passively progress to themselves—recall that diacritical evolution in Section 2.1 is defined for \mathcal{R} verifying $\mathcal{R} \succ \mathcal{R}$, \mathcal{S} .

Weak functions may require \mathcal{R} to make administrative steps (passive progress) before making a computing step (active progress). We reflect this condition in the following notion of *restricted evolution*.

Definition 3.3 Let $(f, g) \in [\mathbb{L} \rightarrow \mathbb{L}]^2$; f *restrictively evolves* to g , written $f \rightsquigarrow^{p|a} g$, if for all $(\mathcal{R}, \mathcal{S}) \in \mathbb{L}^2$, $\mathcal{R} \xrightarrow{p} \mathcal{R}$ and $\mathcal{R} \xrightarrow{a} \mathcal{S}$ imply $f(\mathcal{R}) \xrightarrow{a} g(\mathcal{S})$.

Restricted evolution is a progress relation on $[\mathbb{L} \rightarrow \mathbb{L}]$, and it is preserved by composition under the following hypotheses.

Proposition 3.4 If $f_1 \rightsquigarrow^{p|a} g_1$, and $f_2 \rightsquigarrow^{p|a} g_2$, and $f_2 \xrightarrow{p} f_2$, then $f_1 \circ f_2 \rightsquigarrow^{p|a} g_1 \circ g_2$.

Remark 3.5 The notions of strong and diacritical evolution from our previous work (cf Section 2.1) can be related to the notions of (restricted) evolution of this paper just by unfolding the definitions:

- f strongly evolves to g and h iff $f \xrightarrow{p} g$ and $f \xrightarrow{a} h$.
- If f diacritically evolves to g and h , then $f \xrightarrow{p} g$ and $f \rightsquigarrow^{p|a} h$.

As a result, we reuse the evolution results of our previous papers in the examples about normal-form bisimulation.

We use evolution and restricted evolution to define compatibility for pairs of monotone functions.

Definition 3.6 Let $(s, f) \in [\mathbb{L} \rightarrow \mathbb{L}]^2$. The pair (s, f) is *compatible* if:

$$\begin{array}{ll} s \xrightarrow{p} s & s \xrightarrow{a} f \\ f \xrightarrow{p} f & f \rightsquigarrow^{p|a} f \end{array}$$

The *diacritical companion* is the greatest compatible pair, composed of the *strong* and *weak* companions, written respectively \mathbf{u} and \mathbf{w} :

$$(\mathbf{u}, \mathbf{w}) \triangleq \bigsqcup (s, f). (s, f) \text{ is compatible}$$

A pair (s, f) is compatible if each function is compatible w.r.t. passive progress. The conditions for active progress follow the intuitions hinted before: first, the strong component s evolves towards the weak one, since any up-to technique, not necessary strong ones, can be used after the active progress. Second, the weak component f uses restricted evolution, as f may require the elements of \mathbb{L} to passively progress to themselves before making an active progress.

Example 3.7 In deterministic languages, simulation *up to reduction* [13] allows terms to reduce before being related: we can therefore ignore the intermediary reduction steps and reason in a big-step manner even with a small-step semantics. The definition of the corresponding technique in the λ -calculus is as follows.

$$\frac{e_1 \rightarrow^* e'_1 \quad e_2 \rightarrow^* e'_2 \quad e'_1 \mathcal{R} e'_2}{e_1 \text{ red}(\mathcal{R}) e_2}$$

In [3], we show that $\text{red} \xrightarrow{p} \text{red} \sqcup \text{id}$ and $\text{red} \xrightarrow{a} \text{red} \sqcup \text{id}$, but since $\text{id} \sqsubseteq \text{red}$, we have in fact $\text{red} \xrightarrow{p} \text{red}$ and $\text{red} \xrightarrow{a} \text{red}$. The latter implies that $\text{red} \xrightarrow{p|a} \text{red}$, therefore the pair (red, red) is compatible.

Example 3.8 We give an example of technique for which restricted evolution is needed. We define

$$\frac{e_1 \mathcal{R} e_2 \quad E_1[x] \mathcal{R} E_2[x] \quad x \text{ fresh}}{E_1[e_1] \text{ectx}(\mathcal{R}) E_2[e_2]} \quad \frac{e_1 \mathcal{R} e_2 \quad v_1 x \mathcal{R} v_2 x \quad x \text{ fresh}}{e_1\{v_1/x\} \text{subst}(\mathcal{R}) e_2\{v_2/x\}}$$

The technique ectx allows to factor out different evaluation contexts, as long as these contexts are related when plugged with a fresh variable. Similarly, subst allows to reason up to substitution of related values. We prove in [3] that $\text{subst} \xrightarrow{p} \text{subst}$ and $\text{subst} \xrightarrow{a} \text{ectx} \circ \text{subst}^2$. In contrast, we need restricted evolution to conclude about ectx . We sketch the subcase which illustrates why.

Let \mathcal{R}, \mathcal{S} be such that $\mathcal{R} \xrightarrow{p} \mathcal{R}$ and $\mathcal{R} \xrightarrow{a} \mathcal{S}$, and suppose we have $x v_1 \text{ectx}(\mathcal{R}) E_2[e_2]$ with $x \mathcal{R} e_2$ and $y v_1 \mathcal{R} E_2[y]$ for a fresh y . Since $x v_1$ is an open-stuck term, we have to show that $E_2[e_2]$ also reduces to an open-stuck term. Because $x \mathcal{R} e_2$ and $\mathcal{R} \xrightarrow{p} \mathcal{R}$, there exists v such that $e_2 \rightarrow^* v$ and $x z \mathcal{R} v z$ for a fresh z . Combined with $y v_1 \mathcal{R} E_2[y]$, we get that $x v_1 \text{subst}(\mathcal{R}) E_2[v]$. Because $\mathcal{R} \xrightarrow{a} \mathcal{S}$ and $\text{subst} \xrightarrow{a} \text{ectx} \circ \text{subst}^2$, we know that $E_2[v]$ reduces to an open-stuck term as well. Without restricted evolution, we would have $x v_1 \text{subst}(\mathcal{S}) E_2[v]$, with \mathcal{S} instead of \mathcal{R} , and since we do not have any progress hypothesis about \mathcal{S} , it would not be possible to go further.

More generally, we can show that $\text{ectx} \xrightarrow{p} \text{ectx}$ and $\text{ectx} \xrightarrow{p|a} \text{ectx} \circ \text{subst}^2$ [3]. The pair $(\text{subst}, \text{ectx})$ is not compatible, but we still show it is below the diacritical companion in Section 4. \square

Compatibility for pairs behaves like regular compatibility. In particular, the composition of two compatible pairs produces a compatible pair.

Proposition 3.9

- (i) If (s_1, f_1) and (s_2, f_2) are compatible, then $(s_1 \circ s_2, f_1 \circ f_2)$ is compatible.
- (ii) The pair (id, id) is compatible.
- (iii) The pair (\mathbf{u}, \mathbf{w}) is compatible.
- (iv) Let \mathcal{R} be a simulation, and $\hat{\mathcal{R}}$ the function constant to \mathcal{R} . The pair $(\hat{\mathcal{R}}, \hat{\mathcal{R}})$ is compatible.

Proof. The proofs follow from a straightforward unfolding of the definitions. Item (i) uses Propositions 2.5 and 3.4. \square

Like Pous' companion, \mathbf{u} and \mathbf{w} are idempotent and above id . Besides, similarity can be characterized using \mathbf{w} .

Proposition 3.10 Let $\mathcal{R} \in \mathbb{L}$. The following holds:

- (i) $\text{id} \sqsubseteq \mathbf{u}$, $\text{id} \sqsubseteq \mathbf{w}$,

- (ii) $\mathbf{u} \sqsubseteq \mathbf{u} \circ \mathbf{u}, \quad \mathbf{w} \sqsubseteq \mathbf{w} \circ \mathbf{w},$
 (iii) $\mathbf{u} \circ \mathbf{u} \sqsubseteq \mathbf{u}, \quad \mathbf{w} \circ \mathbf{w} \sqsubseteq \mathbf{w},$
 (iv) $\nu(\overset{\mathbf{p}}{\rightarrow}, \overset{\mathbf{a}}{\rightarrow}) \sqsubseteq \mathbf{u}(\mathcal{R}), \quad \nu(\overset{\mathbf{p}}{\rightarrow}, \overset{\mathbf{a}}{\rightarrow}) \sqsubseteq \mathbf{w}(\mathcal{R}),$
 (v) $\nu(\overset{\mathbf{p}}{\rightarrow}, \overset{\mathbf{a}}{\rightarrow}) = \mathbf{w}(\perp).$

Proof. Item (i) follows from Proposition 3.9, item (ii). Item (ii) is a direct consequence of (i). To prove (iii) we check that $(\mathbf{u} \circ \mathbf{u}, \mathbf{w} \circ \mathbf{w})$ is compatible using the items (i) and (iii) of Proposition 3.9. Because $\nu(\overset{\mathbf{p}}{\rightarrow}, \overset{\mathbf{a}}{\rightarrow})$ is a simulation, we derive (iv) from the item (iv) of Proposition 3.9.

Finally, for (v), we already know that $\nu(\overset{\mathbf{p}}{\rightarrow}, \overset{\mathbf{a}}{\rightarrow}) \sqsubseteq \mathbf{w}(\perp)$ with (iv). For the reverse inequality, we show that $\mathbf{w}(\perp)$ is a simulation. By definition of \perp , we have $\perp \overset{\mathbf{p}}{\rightarrow} \perp$ and $\perp \overset{\mathbf{a}}{\rightarrow} \perp$. Because (\mathbf{u}, \mathbf{w}) is compatible, we get $\mathbf{w}(\perp) \overset{\mathbf{p}}{\rightarrow} \mathbf{w}(\perp)$ and $\mathbf{w}(\perp) \overset{\mathbf{a}}{\rightarrow} \mathbf{w}(\perp)$, as wished. \square

With these properties, we can show the main result of this section, that the diacritical companion is indeed an up-to technique.

Theorem 3.11 *Let $\mathcal{R} \in \mathbb{L}$. If $\mathcal{R} \overset{\mathbf{p}}{\rightarrow} \mathbf{u}(\mathcal{R})$ and $\mathcal{R} \overset{\mathbf{a}}{\rightarrow} \mathbf{w}(\mathcal{R})$ then $\mathcal{R} \sqsubseteq \nu(\overset{\mathbf{p}}{\rightarrow}, \overset{\mathbf{a}}{\rightarrow})$.*

Proof. By the first item of Proposition 3.10, we have $\mathcal{R} \sqsubseteq (\mathbf{w} \circ \mathbf{u})(\mathcal{R})$. We show that $(\mathbf{w} \circ \mathbf{u})(\mathcal{R})$ is a simulation to conclude. From the hypotheses of the theorem, compatibility and idempotence of the companions, we get

$$\mathbf{u}(\mathcal{R}) \overset{\mathbf{p}}{\rightarrow} \mathbf{u}(\mathbf{u}(\mathcal{R})) \sqsubseteq \mathbf{u}(\mathcal{R}) \text{ and } \mathbf{u}(\mathcal{R}) \overset{\mathbf{a}}{\rightarrow} \mathbf{w}(\mathbf{w}(\mathcal{R})) \sqsubseteq \mathbf{w}(\mathcal{R}).$$

Therefore $\mathbf{u}(\mathcal{R}) \overset{\mathbf{p}}{\rightarrow} \mathbf{u}(\mathcal{R})$ and $\mathbf{u}(\mathcal{R}) \overset{\mathbf{a}}{\rightarrow} \mathbf{w}(\mathcal{R})$ because $\overset{\mathbf{p}}{\rightarrow}$ and $\overset{\mathbf{a}}{\rightarrow}$ are progress relations. Using again the compatibility of the diacritical companion, we deduce

$$\mathbf{w}(\mathbf{u}(\mathcal{R})) \overset{\mathbf{p}}{\rightarrow} \mathbf{w}(\mathbf{u}(\mathcal{R})) \text{ and } \mathbf{w}(\mathbf{u}(\mathcal{R})) \overset{\mathbf{a}}{\rightarrow} \mathbf{w}(\mathbf{w}(\mathcal{R})),$$

but $\mathbf{w}(\mathbf{w}(\mathcal{R})) \sqsubseteq \mathbf{w}(\mathcal{R}) = \mathbf{w}(\text{id}(\mathcal{R})) \sqsubseteq \mathbf{w}(\mathbf{u}(\mathcal{R}))$ using the properties of Proposition 3.10. In the end, we obtain $\mathbf{w}(\mathbf{u}(\mathcal{R})) \overset{\mathbf{p}}{\rightarrow} \mathbf{w}(\mathbf{u}(\mathcal{R}))$ and $\mathbf{w}(\mathbf{u}(\mathcal{R})) \overset{\mathbf{a}}{\rightarrow} \mathbf{w}(\mathbf{u}(\mathcal{R}))$, as wished. \square

An immediate consequence is that any pair of monotone functions below the companion is an up-to technique.

Corollary 3.12 *If $(s, f) \sqsubseteq (\mathbf{u}, \mathbf{w})$ then (s, f) is an up-to technique.*

Example 3.13 Let $\theta \triangleq \lambda xy.y (\lambda z.x x y z)$ and $\delta_x \triangleq \lambda y.x (\lambda z.y y z)$; then $\Theta \triangleq \theta \theta$ and $\Delta \triangleq \lambda x.\delta_x \delta_x$ are respectively Turing's and Curry's call-by-value fixed-point combinators. We show that $\theta \theta x$ and $\delta_x \delta_x$ are normal-form similar with a single pair $\mathcal{R} \triangleq \{(\theta \theta x, \delta_x \delta_x)\}$. Indeed, we have $\theta \theta x \rightarrow^* x (\lambda z.\theta \theta x z)$ and $\delta_x \delta_x \rightarrow x (\lambda z.\delta_x \delta_x z)$; we see that the two resulting terms share the common context $x \lambda z.\square z$. Let

$$\frac{e_1 \mathcal{R} e_2}{\lambda x.e_1 \text{ lam}(\mathcal{R}) \lambda x.e_2}$$

Then $\mathcal{R} \xrightarrow{a} (\text{red} \circ \text{ectx} \circ \text{lam} \circ \text{ectx})(\mathcal{R})$. We know that $\text{red} \leq \mathbf{w}$ (Example 3.7) and we show that $\text{ectx} \leq \mathbf{w}$ (Example 4.7) and $\text{lam} \leq \mathbf{w}$ (Example 4.8). Using idempotence of the weak companion, we have $\text{red} \circ \text{ectx} \circ \text{lam} \circ \text{ectx} \sqsubseteq \mathbf{w}^4 \sqsubseteq \mathbf{w}$, so we can conclude with Theorem 3.11. \square

4 Higher-Order Diacritical Companion

Example 3.8 shows that some interesting up-to techniques are not compatible. Yet, we can show they are below the diacritical companion, by turning to higher-order.

4.1 Definition and Properties

The key observation for this section is that the diacritical companion can be defined coinductively, by defining passive and active progresses on pairs of monotone functions. We write \mathbb{L}^\uparrow as a shorthand for the complete lattice $[\mathbb{L} \rightarrow \mathbb{L}] \times [\mathbb{L} \rightarrow \mathbb{L}]$.

Definition 4.1 Let $(s, f), (s', f') \in (\mathbb{L}^\uparrow)^2$. The pair (s, f) *passively evolves* to (s', f') , written $(s, f) \xrightarrow{p} (s', f')$, if $s \xrightarrow{p} s'$ and $f \xrightarrow{p} f'$. It *actively evolves* to (s', f') , written $(s, f) \xrightarrow{a} (s', f')$, if $s \xrightarrow{a} s'$ and $f \xrightarrow{p|a} f'$.

Passive and active evolutions are progress relations over \mathbb{L}^\uparrow . By definition, a compatible pair (s, f) verifies $(s, f) \xrightarrow{p} (s, f)$ and $(s, f) \xrightarrow{a} (s, f)$, i.e., it is a simulation for \xrightarrow{p} and \xrightarrow{a} . Consequently, the corresponding similarity is the diacritical companion, meaning that $(\mathbf{u}, \mathbf{w}) = \nu(\xrightarrow{p}, \xrightarrow{a})$, and we can apply the theory of Section 3 to the lattice \mathbb{L}^\uparrow .

A compatible pair for \mathbb{L}^\uparrow is composed of functions in $[\mathbb{L}^\uparrow \rightarrow \mathbb{L}^\uparrow]$, which we denote with capital letters. The *higher-order diacritical companion*, written (\mathbf{U}, \mathbf{W}) , is the diacritical companion defined w.r.t. \xrightarrow{p} and \xrightarrow{a} . A first result we show by going higher-order is that \mathbf{u} is below \mathbf{w} ; it means that any function below \mathbf{u} can be used as a strong or weak component of an up-to technique. In particular, if $s \sqsubseteq \mathbf{u}$, then (s, s) is an up-to technique.

Proposition 4.2 $\mathbf{u} \sqsubseteq \mathbf{w}$.

Proof. By Theorem 3.11, it is enough to show that $(\mathbf{u}, \mathbf{u}) \xrightarrow{p} \mathbf{U}(\mathbf{u}, \mathbf{u})$ and $(\mathbf{u}, \mathbf{u}) \xrightarrow{a} \mathbf{W}(\mathbf{u}, \mathbf{u})$. For the former, we prove that $(\mathbf{u}, \mathbf{u}) \xrightarrow{p} (\mathbf{u}, \mathbf{u})$, and then use item (i) of Proposition 3.10. But since (\mathbf{u}, \mathbf{w}) is compatible, we have in particular $\mathbf{u} \xrightarrow{p} \mathbf{u}$, from which we deduce $(\mathbf{u}, \mathbf{u}) \xrightarrow{p} (\mathbf{u}, \mathbf{u})$.

To show that $(\mathbf{u}, \mathbf{u}) \xrightarrow{a} \mathbf{W}(\mathbf{u}, \mathbf{u})$, we prove that $(\mathbf{u}, \mathbf{u}) \xrightarrow{a} (\mathbf{u}, \mathbf{w})$; since $(\mathbf{u}, \mathbf{w}) = \nu(\xrightarrow{p}, \xrightarrow{a})$, item (iv) of Proposition 3.10 then gives us $(\mathbf{u}, \mathbf{w}) \sqsubseteq \mathbf{W}(\mathbf{u}, \mathbf{u})$. Because (\mathbf{u}, \mathbf{w}) is compatible, we already have $\mathbf{u} \xrightarrow{a} \mathbf{w}$, which in turn entails $\mathbf{u} \xrightarrow{p|a} \mathbf{w}$. We therefore have $(\mathbf{u}, \mathbf{u}) \xrightarrow{a} (\mathbf{u}, \mathbf{w})$, as wished. \square

Theorem 3.11 is the main technique to show that a pair (s, f) is below (\mathbf{u}, \mathbf{w}) ; we rephrase it to make its application easier in the higher-order setting. First of all, remember that the codomain of \mathbf{U} and \mathbf{W} is a product $\mathbb{L}^\uparrow = [\mathbb{L} \rightarrow \mathbb{L}] \times [\mathbb{L} \rightarrow \mathbb{L}]$,

which we decompose using projections. Let

$$\begin{aligned} \mathbf{U}_{\text{str}} &\triangleq \pi_1 \circ \mathbf{U}, & \mathbf{W}_{\text{str}} &\triangleq \pi_1 \circ \mathbf{W}, \\ \mathbf{U}_{\text{wk}} &\triangleq \pi_2 \circ \mathbf{U}, & \mathbf{W}_{\text{wk}} &\triangleq \pi_2 \circ \mathbf{W}. \end{aligned}$$

Then for all $(s, f) \in \mathbb{L}^\uparrow$, we have $\mathbf{U}(s, f) = (\mathbf{U}_{\text{str}}(s, f), \mathbf{U}_{\text{wk}}(s, f))$ and $\mathbf{W}(s, f) = (\mathbf{W}_{\text{str}}(s, f), \mathbf{W}_{\text{wk}}(s, f))$. With these functions and by unfolding the definitions of $\xrightarrow{\text{p}}$ and $\xrightarrow{\text{a}}$, Theorem 3.11 can be presented as follows.

Corollary 4.3 *Let $(s, f) \in \mathbb{L}^\uparrow$. If*

$$\begin{aligned} s &\xrightarrow{\text{p}} \mathbf{U}_{\text{str}}(s, f) & s &\xrightarrow{\text{a}} \mathbf{W}_{\text{wk}}(s, f) \\ f &\xrightarrow{\text{p}} \mathbf{U}_{\text{wk}}(s, f) & f &\xrightarrow{\text{p|a}} \mathbf{W}_{\text{wk}}(s, f) \end{aligned}$$

then $(s, f) \sqsubseteq (\mathbf{u}, \mathbf{w})$ and (s, f) is an up-to technique.

Following the first-order intuition, \mathbf{U} contains the enhancements allowed after passive evolution, and \mathbf{W} the ones after active evolution. More precisely, \mathbf{U}_{str} contains the enhancements allowed after the evolution of the strong component s , and \mathbf{U}_{wk} those after the weak component f . Due to the asymmetric nature of active evolution, \mathbf{W}_{wk} is used for both components. These functions combine s and f in some way; the question is then which combinations are allowed in the different cases. We provide some answers with the help of the following auxiliary lemma.

Lemma 4.4 *Let $(s, f) \in \mathbb{L}^\uparrow$. We define:*

$$D_1(s, f) \triangleq (s \circ s, s \circ f) \quad D_2(s, f) \triangleq (s \circ s, f \circ f) \quad C(s, f) \triangleq (\perp, f \circ \mathbf{w}).$$

The pairs (D_1, D_2) and (C, C) are compatible.

We relate the higher-order companions to the first-order functions and show how they can be composed. For sake of completeness, we give the properties of \mathbf{W}_{str} even though it is not used in Corollary 4.3.

Proposition 4.5 *Let $(s, f) \in \mathbb{L}^\uparrow$. The following holds:*

$$\begin{aligned} (i) \quad & s \sqsubseteq \mathbf{U}_{\text{str}}(s, f), & s &\sqsubseteq \mathbf{W}_{\text{str}}(s, f), \\ & s \sqsubseteq \mathbf{U}_{\text{wk}}(s, f), & s &\sqsubseteq \mathbf{W}_{\text{wk}}(s, f), \\ (ii) \quad & f \sqsubseteq \mathbf{U}_{\text{wk}}(s, f), & f &\sqsubseteq \mathbf{W}_{\text{wk}}(s, f), \\ (iii) \quad & \mathbf{u} \sqsubseteq \mathbf{U}_{\text{str}}(s, f), & \mathbf{u} &\sqsubseteq \mathbf{W}_{\text{str}}(s, f), \\ & \mathbf{u} \sqsubseteq \mathbf{U}_{\text{wk}}(s, f), & \mathbf{u} &\sqsubseteq \mathbf{W}_{\text{wk}}(s, f), \\ (iv) \quad & \mathbf{w} \sqsubseteq \mathbf{U}_{\text{wk}}(s, f), & \mathbf{w} &\sqsubseteq \mathbf{W}_{\text{wk}}(s, f), \\ (v) \quad & \text{id} \sqsubseteq \mathbf{U}_{\text{str}}(s, f), & \text{id} &\sqsubseteq \mathbf{W}_{\text{str}}(s, f), \\ & \text{id} \sqsubseteq \mathbf{U}_{\text{wk}}(s, f), & \text{id} &\sqsubseteq \mathbf{W}_{\text{wk}}(s, f), \\ (vi) \quad & \mathbf{U}_{\text{str}}(\mathbf{U}(s, f)) \sqsubseteq \mathbf{U}_{\text{str}}(s, f), & \mathbf{W}_{\text{str}}(\mathbf{W}(s, f)) &\sqsubseteq \mathbf{W}_{\text{str}}(s, f), \\ & \mathbf{U}_{\text{wk}}(\mathbf{U}(s, f)) \sqsubseteq \mathbf{U}_{\text{wk}}(s, f), & \mathbf{W}_{\text{wk}}(\mathbf{W}(s, f)) &\sqsubseteq \mathbf{W}_{\text{wk}}(s, f), \end{aligned}$$

$$\begin{aligned}
(vii) \quad & \mathbf{U}_{\text{str}}(s, f) \circ \mathbf{U}_{\text{str}}(s, f) \sqsubseteq \mathbf{U}_{\text{str}}(s, f), & \mathbf{W}_{\text{str}}(s, f) \circ \mathbf{W}_{\text{str}}(s, f) \sqsubseteq \mathbf{W}_{\text{str}}(s, f), \\
& & \mathbf{W}_{\text{wk}}(s, f) \circ \mathbf{W}_{\text{wk}}(s, f) \sqsubseteq \mathbf{W}_{\text{wk}}(s, f), \\
(viii) \quad & \mathbf{U}_{\text{str}}(s, f) \circ \mathbf{U}_{\text{wk}}(s, f) \sqsubseteq \mathbf{U}_{\text{wk}}(s, f), \\
(ix) \quad & \mathbf{U}_{\text{wk}}(s, f) \circ \mathbf{w} \sqsubseteq \mathbf{U}_{\text{wk}}(s, f), \\
(x) \quad & & \mathbf{U}_{\text{str}}(s, f) \sqsubseteq \mathbf{W}_{\text{str}}(s, f), \\
& & \mathbf{U}_{\text{wk}}(s, f) \sqsubseteq \mathbf{W}_{\text{wk}}(s, f), \\
(xi) \quad & \mathbf{U}_{\text{str}}(s, f) \sqsubseteq \mathbf{U}_{\text{wk}}(s, f).
\end{aligned}$$

Proof. By Proposition 3.10, we know that $\text{id} \sqsubseteq \mathbf{U}$ and $\text{id} \sqsubseteq \mathbf{W}$, which imply $(s, f) \sqsubseteq \mathbf{U}(s, f)$ and $(s, f) \sqsubseteq \mathbf{W}(s, f)$, i.e., $s \sqsubseteq \mathbf{U}_{\text{str}}(s, f)$, $f \sqsubseteq \mathbf{U}_{\text{wk}}(s, f)$, $s \sqsubseteq \mathbf{W}_{\text{str}}(s, f)$, and $f \sqsubseteq \mathbf{W}_{\text{wk}}(s, f)$. The remaining inequalities of (i) then follow from items (xi) and (x). Similarly, item (iv) of Proposition 3.10 gives us $(\mathbf{u}, \mathbf{w}) \sqsubseteq \mathbf{U}(s, f)$ and $(\mathbf{u}, \mathbf{w}) \sqsubseteq \mathbf{W}(s, f)$, from which we deduce the inequalities in (iii) and (iv). Item (v) follows from (iii) and the fact that $\text{id} \sqsubseteq \mathbf{u}$, and (vi) is a direct consequence of the idempotence of the (higher-order) companion.

To prove (vii) and (viii), we remark that $D_1 \sqsubseteq \mathbf{U}$ and $D_2 \sqsubseteq \mathbf{W}$ hold by Lemma 4.4. Using idempotence of the higher-order companions, we have the following sequences of inequalities.

$$\begin{aligned}
D_1(\mathbf{U}(s, f)) &\sqsubseteq \mathbf{U}(\mathbf{U}(s, f)) \sqsubseteq \mathbf{U}(s, f) \\
D_2(\mathbf{W}(s, f)) &\sqsubseteq \mathbf{W}(\mathbf{W}(s, f)) \sqsubseteq \mathbf{W}(s, f)
\end{aligned}$$

Unfolding the definition of D_1 and D_2 and taking the appropriate projections give us the inequalities in (vii) and (viii). Similarly, we prove (ix) using compatibility of the pair (C, C) . Item (x) simply recasts Proposition 4.2 in the higher-order setting, and (xi) can be derived from (viii) and (v). \square

Items (v) and (vii) imply that $\mathbf{U}_{\text{str}}(s, f)$, $\mathbf{W}_{\text{str}}(s, f)$, and $\mathbf{W}_{\text{wk}}(s, f)$ are in fact idempotent. This property does not hold for $\mathbf{U}_{\text{wk}}(s, f)$, as shown by the following counterexample.

Example 4.6 We consider normal-form simulation in the λ -calculus as in the previous examples, and define the function f by the following rules, where x , y , and z are any variables.

$$\frac{}{x \, f(\mathcal{R}) \, y} \qquad \frac{x \, \mathcal{R} \, y}{x \, z \, f(\mathcal{R}) \, y \, z}$$

Our goal is to apply Corollary 4.3 to the pair (\perp, f) to eventually show it is an up-to technique. To this end, we study how f evolves w.r.t. \xrightarrow{p} and $\xrightarrow{p|a}$.

We first show that $f \xrightarrow{p} f \circ f$. Let \mathcal{R}, \mathcal{S} such that $\mathcal{R} \xrightarrow{p} \mathcal{S}$; we want $f(\mathcal{R}) \xrightarrow{p} f(f(\mathcal{S}))$. Let $e_1 \, f(\mathcal{R}) \, e_2$; for passive progress, e_1 must be a value, which is possible only if the first rule defining f has been used. Therefore, we have $x \, f(\mathcal{R}) \, y$, and we want $x \, z \, f(f(\mathcal{S})) \, y \, z$ for a fresh z , which holds by definition of f .

We also show that $f \xrightarrow{p|a} f \circ f$. Let \mathcal{R}, \mathcal{S} such that $\mathcal{R} \xrightarrow{p} \mathcal{R}$ and $\mathcal{R} \xrightarrow{a} \mathcal{S}$; we want

$f(\mathcal{R}) \xrightarrow{a} f(f(\mathcal{S}))$. For active progress, only the terms related by the second rule are concerned, i.e., we have $x z f(\mathcal{R}) y z$ with $x \mathcal{R} y$. Because $x z$ and $y z$ are open-stuck terms, we need to show that $x = y$ and $x' z f(f(\mathcal{S})) x' z$ for a fresh x' . Again, the latter holds by definition of f . From $x \mathcal{R} y$ and $\mathcal{R} \xrightarrow{p} \mathcal{R}$, we know that $x z' \mathcal{R} y z'$ for a fresh z' . But these terms are open-stuck, so $\mathcal{R} \xrightarrow{a} \mathcal{S}$ implies that $x = y$, as wished.

As a result, we have $f \xrightarrow{p} f \circ f \sqsubseteq \mathbf{U}_{\text{wk}}(\perp, f) \circ \mathbf{U}_{\text{wk}}(\perp, f)$, and similarly for $\xrightarrow{p|a}$. If $\mathbf{U}_{\text{wk}}(\perp, f) \circ \mathbf{U}_{\text{wk}}(\perp, f) \sqsubseteq \mathbf{U}_{\text{wk}}(\perp, f)$, then we can apply Corollary 4.3, and show that $f \sqsubseteq \mathbf{w}$, which in turn implies that any two variables are normal-form bisimilar, a contradiction. \square

Fortunately, we still have some weaker properties about how \mathbf{U}_{wk} can be composed. The property (viii) shows that any technique allowed after the p -evolution of s can be composed on the left of \mathbf{U}_{wk} , like, for instance, s itself, since we have $s \sqsubseteq \mathbf{U}_{\text{str}}(s, f)$. Similarly, item (ix) allows to compose any technique below \mathbf{w} on the right of \mathbf{U}_{wk} . These properties differ from our previous work, where a weak candidate f should p -evolve towards a function $g_1 \circ f \circ g_2$ such that g_1 and g_2 are built from strong candidate functions (cf. Section 2.1). We still have composition on the left with strong candidates using (viii), but we no longer have composition on the right. Instead, we can compose with techniques that are already proven weak (below \mathbf{w}) using (ix). In Section 4.2, we show that we can recover some composition on the right with strong candidates by changing the hypotheses of Corollary 4.3.

Example 4.7 We remind from Example 3.8 that $\text{subst} \xrightarrow{p} \text{subst}$, $\text{subst} \xrightarrow{a} \text{ectx} \circ \text{subst}^2$, $\text{ectx} \xrightarrow{p} \text{ectx}$, and $\text{ectx} \xrightarrow{p|a} \text{ectx} \circ \text{subst}^2$. From Proposition 4.5, we deduce $\text{subst} \sqsubseteq \mathbf{U}_{\text{str}}(\text{subst}, \text{ectx})$, $\text{ectx} \sqsubseteq \mathbf{U}_{\text{wk}}(\text{subst}, \text{ectx})$, and $\text{subst} \sqsubseteq \mathbf{W}_{\text{wk}}(\text{subst}, \text{ectx})$ by (i), and $\text{ectx} \sqsubseteq \mathbf{W}_{\text{wk}}(\text{subst}, \text{ectx})$ by (ii). Consequently, we have

$$\text{ectx} \circ \text{subst}^2 \sqsubseteq (\mathbf{W}_{\text{wk}}(\text{subst}, \text{ectx}))^3 \sqsubseteq \mathbf{W}_{\text{wk}}(\text{subst}, \text{ectx})$$

by (vi). The pair $(\text{subst}, \text{ectx})$ satisfies the conditions of Corollary 4.3, and is therefore an up-to technique. \square

Example 4.8 We deduce from the previous example that (lam, \perp) is also an up-to technique. Let $\mathcal{R} \xrightarrow{p} \mathcal{S}$, and suppose $\lambda x.e_1 \text{ lam}(\mathcal{R}) \lambda x.e_2$ with $e_1 \mathcal{R} e_2$. For a fresh y , we have $(\lambda x.e_1) y \rightarrow e_1\{y/x\}$, and $(\lambda x.e_2) y \rightarrow e_2\{y/x\}$, so the resulting terms are in $\text{red}(\text{subst}(\mathcal{R}))$, and because $\mathcal{R} \subseteq \mathcal{S}$ by definition of \xrightarrow{p} , we have $\text{lam}(\mathcal{R}) \xrightarrow{p} ((\text{red} \circ \text{subst}) \sqcup \text{lam})(\mathcal{S})$, i.e., $\text{lam} \xrightarrow{p} (\text{red} \circ \text{subst}) \sqcup \text{lam}$. It is enough to conclude, as lam does not actively progress and $(\text{red} \circ \text{subst}) \sqcup \text{lam} \sqsubseteq \mathbf{u}^2 \sqcup \mathbf{U}_{\text{str}}(\text{lam}, \perp) \sqsubseteq \mathbf{U}_{\text{str}}(\text{lam}, \perp)$. \square

Remark 4.9 Example 4.8 shows why progress for normal-form simulation requires $\mathcal{R} \subseteq \mathcal{S}$, as it would not be possible to conclude otherwise. As a result, compatibility in that case is in fact respectfulness [15]. In Pous' work [11], it does not matter to consider either compatibility or respectfulness, as the two notions produce the same companion. We do not know if the same result holds in our framework; Pous' proof relies on the fact that the progress function b is below the companion t . We do not know the relationship between the monotone functions corresponding to \xrightarrow{p} and \xrightarrow{a} ,

and \mathbf{u} and \mathbf{w} .

4.2 Recovering Composition on the Right

As explained before, the properties we have so far for \mathbf{U}_{wk} do not allow to compose strong candidates on the right, while our previous theory permits it. We show that we can recover such a property by only slightly strengthening the hypotheses of Corollary 4.3.

We first observe that Proposition 4.5 does not relate in any way a weak candidate f and $\mathbf{U}_{\text{str}}(s, f)$. As a result, given h, s, f , and a proof of $h \sqsubseteq \mathbf{U}_{\text{str}}(s, f)$ which uses solely the inequalities of Proposition 4.5, we can easily use the same proof to show that $h \sqsubseteq \mathbf{U}_{\text{str}}(s, \perp)$. The function $\mathbf{U}_{\text{str}}(s, \perp)$ is interesting because it does not mention any weak candidate, so we can study how it evolves more easily.

Lemma 4.10 *Let $(s, f) \in \mathbb{L}^\uparrow$. If $s \xrightarrow{p} \mathbf{U}_{\text{str}}(s, \perp)$ then $\mathbf{U}_{\text{str}}(s, \perp) \xrightarrow{p} \mathbf{U}_{\text{str}}(s, \perp)$. If $s \xrightarrow{a} \mathbf{W}_{\text{wk}}(s, f)$ then $\mathbf{U}_{\text{str}}(s, \perp) \xrightarrow{\text{pla}} \mathbf{W}_{\text{wk}}(s, f)$.*

Proof. By definition of \perp and progress, we have, respectively, $\perp \xrightarrow{p} \mathbf{U}_{\text{wk}}(s, \perp)$ and $\perp \xrightarrow{\text{pla}} \mathbf{U}_{\text{wk}}(s, f)$, and therefore $(s, \perp) \xrightarrow{p} \mathbf{U}(s, \perp)$ and $(s, \perp) \xrightarrow{a} \mathbf{W}(s, f)$. Because (\mathbf{U}, \mathbf{W}) is compatible, we deduce $\mathbf{U}(s, \perp) \xrightarrow{p} \mathbf{U}(\mathbf{U}(s, \perp))$ and $\mathbf{U}(s, \perp) \xrightarrow{a} \mathbf{W}(\mathbf{W}(s, f))$. We conclude by idempotence of \mathbf{U} and \mathbf{W} and with the appropriate projections. \square

To achieve the goal of this section, we define a function \mathbf{U}_{rc} which contains \mathbf{U}_{wk} and allows for composition on the right with $\mathbf{U}_{\text{str}}(s, \perp)$. Let $(s, f) \in \mathbb{L}^\uparrow$; we define \mathbf{U}_{rc} as the smallest function verifying the following properties.

- (i) $f \sqsubseteq \mathbf{U}_{\text{rc}}(s, f)$,
- (ii) $\mathbf{U}_{\text{wk}}(s, \mathbf{U}_{\text{rc}}(s, f)) \sqsubseteq \mathbf{U}_{\text{rc}}(s, f)$,
- (iii) $\mathbf{U}_{\text{rc}}(s, f) \circ \mathbf{U}_{\text{str}}(s, \perp) \sqsubseteq \mathbf{U}_{\text{rc}}(s, f)$.

We show that \mathbf{U}_{rc} indeed contains \mathbf{U}_{wk} , from which we deduce some properties.

Proposition 4.11 *Let $(s, f) \in \mathbb{L}^\uparrow$. The following holds:*

- (i) $\mathbf{U}_{\text{wk}}(s, f) \sqsubseteq \mathbf{U}_{\text{rc}}(s, f)$,
- (ii) $s \sqsubseteq \mathbf{U}_{\text{rc}}(s, f)$,
- (iii) $\mathbf{U}_{\text{str}}(s, f) \sqsubseteq \mathbf{U}_{\text{rc}}(s, f)$.
- (iv) $\mathbf{U}_{\text{str}}(s, f) \circ \mathbf{U}_{\text{rc}}(s, f) \sqsubseteq \mathbf{U}_{\text{rc}}(s, f)$.

Proof. Because \mathbf{U}_{wk} is monotone and by definition of \mathbf{U}_{rc} , we have $\mathbf{U}_{\text{wk}}(s, f) \sqsubseteq \mathbf{U}_{\text{wk}}(s, \mathbf{U}_{\text{rc}}(s, f)) \sqsubseteq \mathbf{U}_{\text{rc}}(s, f)$. The second and third items are then direct consequences of the first one and Proposition 4.5. For the last one,

$$\begin{aligned}
 \mathbf{U}_{\text{str}}(s, f) \circ \mathbf{U}_{\text{rc}}(s, f) &\sqsubseteq \mathbf{U}_{\text{str}}(s, \mathbf{U}_{\text{rc}}(s, f)) \circ \mathbf{U}_{\text{rc}}(s, f) && \mathbf{U}_{\text{str}} \text{ monotone; definition } \mathbf{U}_{\text{rc}} \\
 &\sqsubseteq \mathbf{U}_{\text{str}}(s, \mathbf{U}_{\text{rc}}(s, f)) \circ \mathbf{U}_{\text{wk}}(s, \mathbf{U}_{\text{rc}}(s, f)) && \text{Prop 4.5, item (ii)} \\
 &\sqsubseteq \mathbf{U}_{\text{wk}}(s, \mathbf{U}_{\text{rc}}(s, f)) && \text{Prop 4.5, item (viii)} \\
 &\sqsubseteq \mathbf{U}_{\text{rc}}(s, f) && \text{definition of } \mathbf{U}_{\text{rc}}.
 \end{aligned}$$

□

We modify Corollary 4.3 to use \mathbf{U}_{rc} instead of \mathbf{U}_{wk} . The other change we need is the hypothesis about the \mathbf{p} -evolution of \mathbf{s} , which is now towards $\mathbf{U}_{str}(\mathbf{s}, \perp)$ instead of $\mathbf{U}_{str}(\mathbf{s}, \mathbf{f})$. As explained at the beginning of the section, the change does not matter much when the evolution proof is based on the inequalities of Proposition 4.5.

Theorem 4.12 *Let $(\mathbf{s}, \mathbf{f}) \in \mathbb{L}^\uparrow$. If*

$$\begin{array}{ll} \mathbf{s} \xrightarrow{\mathbf{p}} \mathbf{U}_{str}(\mathbf{s}, \perp), & \mathbf{s} \xrightarrow{\mathbf{a}} \mathbf{W}_{wk}(\mathbf{s}, \mathbf{f}) \\ \mathbf{f} \xrightarrow{\mathbf{p}} \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f}), & \mathbf{f} \xrightarrow{\mathbf{p|a}} \mathbf{W}_{wk}(\mathbf{s}, \mathbf{f}). \end{array}$$

then $(\mathbf{s}, \mathbf{f}) \sqsubseteq (\mathbf{u}, \mathbf{w})$, and (\mathbf{s}, \mathbf{f}) is an up-to technique.

Proof. We prove that $(\mathbf{s}, \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})) \sqsubseteq (\mathbf{u}, \mathbf{w})$ using Corollary 4.3. For \mathbf{s} , we need $\mathbf{s} \xrightarrow{\mathbf{p}} \mathbf{U}_{str}(\mathbf{s}, \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f}))$ and $\mathbf{s} \xrightarrow{\mathbf{a}} \mathbf{W}_{wk}(\mathbf{s}, \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f}))$, which can be derived directly from the hypotheses of the theorem, since $\perp \sqsubseteq \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})$, $\mathbf{f} \sqsubseteq \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})$, and \mathbf{U}_{str} and \mathbf{W}_{wk} are monotone,

We show that $\mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})$ also respects the hypotheses of Corollary 4.3, by proving first that $\mathbf{s} \xrightarrow{\mathbf{p}} \mathbf{U}_{str}(\mathbf{s}, \perp)$ and $\mathbf{f} \xrightarrow{\mathbf{p}} \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})$ imply $\mathbf{U}_{rc}(\mathbf{s}, \mathbf{f}) \xrightarrow{\mathbf{p}} \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})$. We proceed by induction on the definition of \mathbf{U}_{rc} . For the base case, we have to show that $\mathbf{f} \xrightarrow{\mathbf{p}} \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})$, which is an assumption.

For the inductive cases, let \mathbf{h} such that $\mathbf{h} \xrightarrow{\mathbf{p}} \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})$; we show $\mathbf{U}_{rc}(\mathbf{s}, \mathbf{h}) \xrightarrow{\mathbf{p}} \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})$ and $\mathbf{h} \circ \mathbf{U}_{str}(\mathbf{s}, \perp) \xrightarrow{\mathbf{p}} \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})$. For the first one, we have $(\mathbf{s}, \mathbf{h}) \xrightarrow{\mathbf{p}} (\mathbf{U}_{str}(\mathbf{s}, \perp), \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f}))$ by the assumptions. Since (\mathbf{U}, \mathbf{W}) is compatible, we have in particular $\mathbf{U}(\mathbf{s}, \mathbf{h}) \xrightarrow{\mathbf{p}} \mathbf{U}(\mathbf{U}_{str}(\mathbf{s}, \perp), \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f}))$, which implies $\mathbf{U}_{wk}(\mathbf{s}, \mathbf{h}) \xrightarrow{\mathbf{p}} \mathbf{U}_{wk}(\mathbf{U}_{str}(\mathbf{s}, \perp), \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f}))$. But

$$\begin{aligned} \mathbf{U}_{wk}(\mathbf{U}_{str}(\mathbf{s}, \perp), \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})) &\sqsubseteq \mathbf{U}_{wk}(\mathbf{U}_{str}(\mathbf{s}, \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})), \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})) && \perp \sqsubseteq \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f}) \\ &\sqsubseteq \mathbf{U}_{wk}(\mathbf{U}_{str}(\mathbf{s}, \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})), \mathbf{U}_{wk}(\mathbf{s}, \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f}))) && \text{Prop 4.5, item (ii)} \\ &\sqsubseteq \mathbf{U}_{wk}(\mathbf{U}(\mathbf{s}, \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f}))) && \text{definition } \mathbf{U}_{str}, \mathbf{U}_{wk} \\ &\sqsubseteq \mathbf{U}_{wk}(\mathbf{s}, \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})) && \text{Prop 4.5, item (vi)} \\ &\sqsubseteq \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f}) && \text{definition } \mathbf{U}_{rc}. \end{aligned}$$

Therefore, we have $\mathbf{U}_{wk}(\mathbf{s}, \mathbf{h}) \xrightarrow{\mathbf{p}} \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})$, as wished.

We now prove $\mathbf{h} \circ \mathbf{U}_{str}(\mathbf{s}, \perp) \xrightarrow{\mathbf{p}} \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})$ with Proposition 3.4. Indeed $\mathbf{h} \xrightarrow{\mathbf{p}} \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})$ holds by assumption, and $\mathbf{U}_{str}(\mathbf{s}, \perp) \xrightarrow{\mathbf{p}} \mathbf{U}_{str}(\mathbf{s}, \perp)$ from Lemma 4.10, from which we deduce $\mathbf{U}_{str}(\mathbf{s}, \perp) \xrightarrow{\mathbf{p}} \mathbf{U}_{str}(\mathbf{s}, \perp)$. As a result, we get $\mathbf{h} \circ \mathbf{U}_{str}(\mathbf{s}, \perp) \xrightarrow{\mathbf{p}} \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f}) \circ \mathbf{U}_{str}(\mathbf{s}, \perp)$, which is enough since $\mathbf{U}_{rc}(\mathbf{s}, \mathbf{f}) \circ \mathbf{U}_{str}(\mathbf{s}, \perp) \sqsubseteq \mathbf{U}_{rc}(\mathbf{s}, \mathbf{f})$ by definition of \mathbf{U}_{rc} .

Similarly, we can show that $\mathbf{s} \xrightarrow{\mathbf{p}} \mathbf{U}_{str}(\mathbf{s}, \perp)$, $\mathbf{s} \xrightarrow{\mathbf{a}} \mathbf{W}_{wk}(\mathbf{s}, \mathbf{f})$, and $\mathbf{f} \xrightarrow{\mathbf{p|a}} \mathbf{W}_{wk}(\mathbf{s}, \mathbf{f})$ imply $\mathbf{U}_{rc}(\mathbf{s}, \mathbf{f}) \xrightarrow{\mathbf{p|a}} \mathbf{W}_{wk}(\mathbf{s}, \mathbf{f})$. In the end, $\mathbf{U}_{rc}(\mathbf{f}, \mathbf{s})$ satisfies the conditions of Corollary 4.3, as wished. □

As an application of Theorem 4.12, we show that the present theory subsumes our previous work. Let \mathbb{F} be a set of monotone functions compatible as defined in Section 2.1, with \mathbb{S} as its largest subset of strong functions. We prove that for

any $s \in \mathbb{S}$ and $f \in \mathbb{F}$, we have $(s, f) \sqsubseteq (\mathbf{u}, \mathbf{w})$. To do so, given a set \mathbb{X} of monotone functions, we consider $\text{set}_{\mathbb{X}} \triangleq \bigsqcup f.f \in \mathbb{X}$ as well as the largest function built from \mathbb{X} using composition and union, written $\text{set}_{\mathbb{X}}^{\omega}$, which can be defined inductively as follows.

- $\text{id} \sqsubseteq \text{set}_{\mathbb{X}}^{\omega}$;
- if $f \in \mathbb{X}$, then $f \circ \text{set}_{\mathbb{X}}^{\omega} \sqsubseteq \text{set}_{\mathbb{X}}^{\omega}$, $\text{set}_{\mathbb{X}}^{\omega} \circ f \sqsubseteq \text{set}_{\mathbb{X}}^{\omega}$, $f \sqcup \text{set}_{\mathbb{X}}^{\omega} \sqsubseteq \text{set}_{\mathbb{X}}^{\omega}$, and $\text{set}_{\mathbb{X}}^{\omega} \sqcup f \sqsubseteq \text{set}_{\mathbb{X}}^{\omega}$.

The definition of compatible set (Section 2.1) and Remark 3.5 then imply $s \xrightarrow{p} \text{set}_{\mathbb{S}}^{\omega}$, $s \xrightarrow{a} \text{set}_{\mathbb{F}}^{\omega}$, $f \xrightarrow{p} \text{set}_{\mathbb{S}}^{\omega} \circ f \circ \text{set}_{\mathbb{S}}^{\omega}$, and $f \xrightarrow{p|a} \text{set}_{\mathbb{F}}^{\omega}$, and therefore $\text{set}_{\mathbb{S}} \xrightarrow{p} \text{set}_{\mathbb{S}}^{\omega}$, $\text{set}_{\mathbb{S}} \xrightarrow{a} \text{set}_{\mathbb{F}}^{\omega}$, $\text{set}_{\mathbb{F}} \xrightarrow{p} \text{set}_{\mathbb{S}}^{\omega} \circ \text{set}_{\mathbb{F}} \circ \text{set}_{\mathbb{S}}^{\omega}$, and $\text{set}_{\mathbb{F}} \xrightarrow{p|a} \text{set}_{\mathbb{F}}^{\omega}$. It is easy to show by induction that $\text{set}_{\mathbb{S}}^{\omega} \sqsubseteq \mathbf{U}_{\text{str}}(\text{set}_{\mathbb{S}}, \perp)$ and $\text{set}_{\mathbb{F}}^{\omega} \sqsubseteq \mathbf{W}_{\text{wk}}(\text{set}_{\mathbb{S}}, \text{set}_{\mathbb{F}})$. As a result, we have $\text{set}_{\mathbb{S}}^{\omega} \circ \text{set}_{\mathbb{F}} \circ \text{set}_{\mathbb{S}}^{\omega} \sqsubseteq \mathbf{U}_{\text{str}}(\text{set}_{\mathbb{S}}, \perp) \circ \mathbf{U}_{\text{wk}}(\text{set}_{\mathbb{S}}, \text{set}_{\mathbb{F}}) \circ \mathbf{U}_{\text{str}}(\text{set}_{\mathbb{S}}, \perp) \sqsubseteq \mathbf{U}_{\text{rc}}(\text{set}_{\mathbb{S}}, \text{set}_{\mathbb{F}})$. The pair $(\text{set}_{\mathbb{S}}, \text{set}_{\mathbb{F}})$ satisfies the conditions of Theorem 4.12, and is therefore below (\mathbf{u}, \mathbf{w}) .

4.3 Symmetry

So far, the results and examples are phrased in terms of simulation and similarity. As in Pous [11], we show that assuming symmetry of the underlying progress relations, we can go from the similarity to the bisimilarity. Let $i \in [\mathbb{L} \rightarrow \mathbb{L}]$ be an involution, i.e., $i \circ i = \text{id}$. An element $\mathcal{R} \in \mathbb{L}$ is symmetric if $i(\mathcal{R}) = \mathcal{R}$, and a function $f \in [\mathbb{L} \rightarrow \mathbb{L}]$ is symmetric if $f \circ i = i \circ f$. For binary relations, i is the usual inverse \cdot^{-1} .

Let $\xrightarrow{b} \in \{\xrightarrow{p}, \xrightarrow{a}\}$; suppose there exists $\xrightarrow{\bar{b}}$ such that $\xrightarrow{b} = \xrightarrow{\bar{b}} \wedge i \circ \xrightarrow{\bar{b}} \circ i$. Then for all $(b, b') \in \{p, a\}^2$, we also have $\xrightarrow{b} = \xrightarrow{\bar{b}} \wedge i \circ \xrightarrow{\bar{b}} \circ i$ and $\xrightarrow{b|b'} = \xrightarrow{\bar{b}|\bar{b}'} \wedge i \circ \xrightarrow{\bar{b}|\bar{b}'} \circ i$. We can relate progresses and evolutions as follows.

Proposition 4.13 *Let $(\mathcal{R}, \mathcal{S}) \in \mathbb{L}^2$ such that \mathcal{R} is symmetric, and $f \in \{\mathbf{u}, \mathbf{w}\}$. Then $\mathcal{R} \xrightarrow{b} f(\mathcal{S})$ iff $\mathcal{R} \xrightarrow{\bar{b}} f(\mathcal{S})$.*

Let $(s, f, g) \in (\mathbb{L}^{\uparrow})^3$ such that g is symmetric, and $F \in \{\mathbf{U}_{\text{str}}, \mathbf{U}_{\text{wk}}, \mathbf{U}_{\text{rc}}, \mathbf{W}_{\text{str}}, \mathbf{W}_{\text{wk}}\}$. Then:

$$(i) \quad g \xrightarrow{b} F(s, f) \text{ iff } g \xrightarrow{\bar{b}} F(s, f);$$

$$(ii) \quad g \xrightarrow{b|b'} F(s, f) \text{ iff } g \xrightarrow{\bar{b}|\bar{b}'} F(s, f).$$

Proof. The left-to-right implications are by hypothesis on \xrightarrow{b} , so we focus on the right-to-left ones. First, we show that (i, \perp) is compatible. Indeed, let $\mathcal{R} \xrightarrow{b} \mathcal{S}$; then $\mathcal{R} \xrightarrow{\bar{b}} \mathcal{S}$ and $i(\mathcal{R}) \xrightarrow{\bar{b}} i(\mathcal{S})$. Since i is involutive, we have $i(i(\mathcal{R})) \xrightarrow{\bar{b}} i(i(\mathcal{S}))$ and $i(\mathcal{R}) \xrightarrow{\bar{b}} i(\mathcal{S})$, i.e., $i(\mathcal{R}) \xrightarrow{b} i(\mathcal{S})$, as wished. Therefore, $i \sqsubseteq \mathbf{u} \sqsubseteq \mathbf{w}$.

For the first-order property, from $i \sqsubseteq f$, $i \circ i = \text{id}$, and idempotence of the companions, we deduce $i \circ f = f$. Therefore, $\mathcal{R} \xrightarrow{\bar{b}} f(\mathcal{S})$ implies $\mathcal{R} \xrightarrow{\bar{b}} i(f(\mathcal{S}))$, i.e., $i(\mathcal{R}) \xrightarrow{\bar{b}} i(f(\mathcal{S}))$ since \mathcal{R} is symmetric. In the end, we have $\mathcal{R} \xrightarrow{b} f(\mathcal{S})$, as expected.

For the higher-order properties, we first notice that $i \circ F(s, f) = F(s, f) = F(s, f) \circ i$. If $F \in \{\mathbf{U}_{\text{str}}, \mathbf{W}_{\text{str}}, \mathbf{W}_{\text{wk}}\}$, then $i \circ F(s, f) \sqsubseteq \mathbf{u} \circ F(s, f) \sqsubseteq F(s, f) \circ F(s, f) \sqsubseteq F(s, f)$ using items (iii) and (vi) of Proposition 4.5. From $i \circ F(s, f) \sqsubseteq F(s, f)$, we get

$i \circ i \circ F(s, f) \sqsubseteq i \circ F(s, f)$, i.e., $F(s, f) \sqsubseteq i \circ F(s, f)$, and therefore $i \circ F(s, f) = F(s, f)$. The proof for $F(s, f) \circ i = F(s, f)$ is similar. If $F = \mathbf{U}_{\text{wk}}$, the proof is the same using the items (viii) and (ix) of Proposition 4.5, and for $F = \mathbf{U}_{\text{rc}}$, use item (iv) of Proposition 4.11 and item (iii) from the definition of \mathbf{U}_{rc} .

We now prove item (i) of Proposition 4.13, the proof for (ii) being similar. Let $\mathcal{R} \xrightarrow{b} \mathcal{S}$; then $\mathcal{R} \xrightarrow{\bar{b}} \mathcal{S}$ and $i(\mathcal{R}) \xrightarrow{\bar{b}} i(\mathcal{S})$. Because $g \xrightarrow{\bar{b}} F(s, f)$, it implies $g(\mathcal{R}) \xrightarrow{\bar{b}} F(s, f)(\mathcal{S})$ and $g(i(\mathcal{R})) \xrightarrow{\bar{b}} F(s, f)(i(\mathcal{S}))$. Because g is symmetric and $F(s, f) \circ i = i \circ F(s, f)$, the last progress can be rewritten as $i(g(\mathcal{R})) \xrightarrow{\bar{b}} i(F(s, f)(\mathcal{S}))$, meaning that $g(\mathcal{R}) \xrightarrow{b} F(s, f)(\mathcal{S})$ holds, and $g \xrightarrow{b} F(s, f)$. \square

5 Conclusion

In this paper, we propose, through the diacritical companion, a sufficient criterion to show in the presence of passive and active progresses that a function is an up-to technique. The key change compared to our previous work is the decomposition of an up-to technique into its strong and weak components, allowing a definition of active evolution where a strong component may evolve to a weak one. The diacritical companion enjoys most of the properties of Pous' companion, and in particular, when considered in the higher-order setting, it provides a proof technique to show that a pair of functions is below the first-order diacritical companion. A natural future work is then to investigate if the subsequent characterizations of Pous' companion [9,16,12,2] scale to the diacritical one.

Acknowledgement

We thank the anonymous reviewers for their helpful comments on the presentation of this work. This work was partially supported by COST Action EUTypes CA15123.

References

- [1] A. Aristizábal, D. Biernacki, S. Lenglet, and P. Polesiuk. Environmental Bisimulations for Delimited-Control Operators with Dynamic Prompt Generation. *Logical Methods in Computer Science*, 13(3), 2017.
- [2] H. Basold, D. Pous, and J. Rot. Monoidal company for accessible functors. In F. Bonchi and B. König, editors, *7th Conference on Algebra and Coalgebra in Computer Science, CALCO 2017*, volume 72 of *LIPICs*, pages 5:1–5:16, Ljubljana, Slovenia, June 2017. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.
- [3] D. Biernacki, S. Lenglet, and P. Polesiuk. Proving soundness of extensional normal-form bisimilarities. In A. Silva, editor, *Proceedings of the 33th Annual Conference on Mathematical Foundations of Programming Semantics (MFPS XXXIII)*, volume 336 of *Electronic Notes in Theoretical Computer Science*, pages 41–56, Ljubljana, Slovenia, June 2017.
- [4] F. Bonchi, D. Petrisan, D. Pous, and J. Rot. Coinduction up-to in a fibrational setting. In T. A. Henzinger and D. Miller, editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14*, pages 20:1–20:9. ACM, July 2014.
- [5] N. A. Danielsson. Up-to techniques using sized types. *PACMPL*, 2(POPL):43:1–43:28, 2018.
- [6] C. Hur, G. Neis, D. Dreyer, and V. Vafeiadis. The power of parameterization in coinductive proof. In R. Giacobazzi and R. Cousot, editors, *The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '13*, pages 193–206, Rome, Italy, Jan. 2013. ACM.

- [7] S. B. Lassen. Eager normal form bisimulation. In P. Panangaden, editor, *LICS'05*, pages 345–354, Chicago, IL, June 2005. IEEE Computer Society Press.
- [8] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
- [9] J. Parrow and T. Weber. The largest respectful function. *Logical Methods in Computer Science*, 12(2), 2016.
- [10] D. Pous. Complete lattices and up-to techniques. In Z. Shao, editor, *APLAS'07*, number 4807 in LNCS, pages 351–366, Singapore, Dec. 2007. Springer-Verlag.
- [11] D. Pous. Coinduction all the way up. In M. Grohe, E. Koskinen, and N. Shankar, editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16*, pages 307–316, New York, NY, USA, July 2016. ACM.
- [12] D. Pous and J. Rot. Companions, codensity and causality. In J. Esparza and A. S. Murawski, editors, *Foundations of Software Science and Computation Structures - 20th International Conference, FOSSACS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017*, volume 10203 of *Lecture Notes in Computer Science*, pages 106–123, Uppsala, Sweden, Apr. 2017.
- [13] D. Pous and D. Sangiorgi. Enhancements of the bisimulation proof method. In D. Sangiorgi and J. Rutten, editors, *Advanced Topics in Bisimulation and Coinduction*, chapter 6, pages 233–289. Cambridge University Press, 2011.
- [14] J. J. M. M. Rutten. A coinductive calculus of streams. *Mathematical Structures in Computer Science*, 15(1):93–147, 2005.
- [15] D. Sangiorgi. On the bisimulation proof method. *Mathematical Structures in Computer Science*, 8(5):447–479, Oct. 1998.
- [16] S. Schäfer and G. Smolka. Tower induction and up-to techniques for CCS with fixed points. In P. Höfner, D. Pous, and G. Struth, editors, *Relational and Algebraic Methods in Computer Science - 16th International Conference, RAMiCS 2017, Lyon, France, May 15-18, 2017, Proceedings*, volume 10226 of *Lecture Notes in Computer Science*, pages 274–289, 2017.