Zig-zag and Extraction Families in Non-duplicating Stable Deterministic Residual Structures

Zurab Khasidashvili and John Glauert

School of Information Systems, UEA
Norwich NR4 7TJ England
zurab@sys.uea.ac.uk, jrwg@sys.uea.ac.uk *
Tel: +44 1603 592671, Fax: +44 1603 593344

Abstract. In a previous paper, we introduced *Stable Deterministic Residual Structures* (SDRSs), Abstract Reduction Systems with an axiomatized residual relation which model orthogonal term and graph rewriting systems, and *Deterministic Family Structures* (DFSs), which are SDRSs with an axiomatized notion of *redex-family* to capture known *sharing* concepts in the λ-calculus and other orthogonal rewrite systems. In this paper, we start to investigate ways of constructing DFSs from SDRSs. This is interesting for at least the following two purposes: (1) to develop an *algebraic* theory of sharing for conflict-free rewrite and transition systems in order understand what properties a sharing concept must possess in order to imply a reasonable theory of optimal evaluation, and (2) to give an *Event Structure* style semantics to conflict-free rewrite and transition systems with *erasure*. As a first step, which is already quite complicated, we only consider non-duplicating systems, and show that every non-duplicating SDRS is already a DFS if the *zig-zag* is taken for the family relation. (Zig-zag is simply the reflexive and transitive closure of the residual relation on redexes with histories.) To achieve this, we needed to develop an abstract *extraction* procedure, which was thought to require the tree structure of terms, and to show that the family concepts defined via zig-zag and via extraction yield the same relation. As a side result, we get a Prime Event Structure semantics for non-duplicating SDRSs. Various forms of conflict-free Graph Rewriting Systems (or Graph Grammars) fall in the category SDRSs.

1 Introduction

In order to achieve optimal evaluation of λ-terms, Lévy introduced a notion of *redex-family* to capture the concept of redexes of the 'same origin', hoping that it would be possible to mimic reductions contracting whole families in multi-steps by reduction of some graph representation in which every multi-step would be represented by contraction of a single redex [Lév78, Lév80]. There was no other way – Barendregt et al [BBKV76] showed that there does not exist a one-step optimal recursive β-reduction strategy on λ-terms. Such an implementation has indeed been achieved by Lamping [Lam90] and Kathail [Kat90], reviving interest in optimal graph reduction. Maranget [Mar91] generalized Lévy's optimality theory to Orthogonal Term Rewriting Systems (OTRSs), Gonthier et al [GAL92] simplified Lamping's technique, and Asperti and Laneve generalized both Lévy's optimality theory and Gonthier's implementation of it to Interaction Systems, which cover most of the languages with a constructor-destructor discipline [AsLa93, AsLa96]. Recently, the optimality theory has been generalized to the whole class of orthogonal Higher-order Rewrite Systems (HORSs) [Oos96].

Lévy introduced the family concept in three different ways: via a suitable notion of *labelling*, via *extraction*, and as *zig-zag*. In each definition, a family is a class of objects of the form Pv, where P is a finite reduction starting from a term t and ending in s, and v is a redex in s. Here P is called the *history* of Pv; all histories of redexes in the same family are co-initial, i.e., start at the same term t. In the labelling definition of families, the initial term t gets an *initial* labelling, and labels grow along the reduction. Two redexes with co-initial histories are declared to be in the same family if they have the same label. The extraction process

* This work was supported by the Engineering and Physical Sciences Research Council of Great Britain under grant GR/H 41300

starting from a redex Pv consists of elimination from P of all redexes 'not contributing' to v, and results in a reduction $P'v'$ such that P' is standard (therefore unique in its permutation-equivalence class), and Pv is a residual of $P'v'$, i.e., v is a residual of v' under some reduction from the final term of P' to that of P. Two redexes (with histories) are defined to be in the same extraction-family if the extraction process yields the same result for both redexes. Finally, the zig-zag relation is defined simply as the transitive and reflexive closure of the residual relation on redexes with histories. Clearly, all the above family concepts are equivalence relations, and Lévy showed that they all yield the same concept, in the λ-calculus.

The same holds for all orthogonal HORSs R, if all three family concepts are defined in the refinement of R which decomposes every original R-step into first-order or TRS-step and a number of substitution steps [Oos96]. However, the zig-zag family can be defined directly in R, and this yields a different family concept [AsLa93]. Independently and much earlier, Kennaway and Sleep [KeSl89] defined their concept of labelling for orthogonal Combinatory Reduction Systems (CRSs), improving Klop's original labelling system for CRSs [Klo80], which cover orthogonal TRSs and Interaction Systems, and their labelling is different from both Maranget's labelling for OTRSs [Mar91] and Asperti-Laneve's labelling for Interaction Systems [AsLa93].

This variety of family concepts, and development of alternative graph rewriting algorithms for optimal implementation of orthogonal rewriting systems, such as Term Graph Rewriting [KKSV93], Jungle rewriting [HP91], DAG (Directed Acyclic Graph) rewriting [Mar91], and many others (in particular, covering cyclic graph reduction as well), inspired by Wadsworth's original work on graph-based implementation of the λ-calculus [Wad71], created the need to develop an abstract notion of family general enough to cover all the existing notions, and deep enough to enable proof of normalization and optimality results. Such structures were indeed introduced by the authors of this paper in [GlKh96] as *Deterministic Family Structures* (DFSs). This became possible also due to recent developments of abstract reduction systems with axiomatized residual relation, such as *Concurrent Transition Systems* (CTSs) of Stark [Sta89] and *Abstract Reduction Systems* of Gonthier et al [GLM92].

Our DFSs are defined as *Deterministic Residual Structures* (DRSs) with axiomatized family relation. DRSs, in turn, are Abstract Reduction Systems with axiomatized residual relation, similar to CTSs [Sta89] and ARSs [GLM92], but with the difference that, unlike CTSs, the residual relation can be duplicating in DRSs, and unlike ARSs, there is no nesting relation on redexes defined or axiomatized in DRSs. (Therefore, DRSs cover more rewrite and transition systems than conflict-free CTSs or ARSs.) CTSs have successfully been used to give semantics to machine networks, while ARSs have been used to study more syntactic properties of orthogonal rewrite systems, such as standardization. Despite its highly abstract nature, a counterpart of Berry's *stability* property [Ber79] enables one to prove the normalization theorem for all DRSs, and not only w.r.t. normal forms, but in general for *(regular) stable sets* of '(partial) results'; all interesting sets of final terms, such as head-normal forms, weak head normal forms, etc, fall in the category of stable term-sets [GlKh96]. Moreover, a *discrete* theory of normalization can also be developed in Berry-stable DRSs, enabling one to construct reductions permutation-equivalent to a particular finite or infinite reduction, and to prove a version of the Standardization Theorem [KhGl96]. Further, as already mentioned, in DFSs one can prove the optimality theorem, and DFSs can be interpreted as (*deterministic* or *conflict-free*) Prime Event Structures (DPESs) [NPW81, Win88].

In this paper, we study the possibility of defining a family relation in stable DRSs. We will only consider non-duplicating DRSs, because of its particularly important semantic applications, and will show that the zig-zag relation is a family relation in the sense that it satisfies the family-axioms of DFSs. So every non-duplicating stable DRSs is in fact a DFS with the zig-zag as the family relation. This is achieved by defining an abstract extraction procedure and showing that zig-zag coincides with the extraction-family relation. For the extraction-family concept, checking the DFS family axioms are easy. Since families in DFSs, ordered by the contribution relation, form DPESs, our construction yields a translation of stable determinate CTSs into deterministic stable Event structures, linking two widely accepted (operational and set-theoretic) models of computation.

The technical contribution is the simplicity of our construction which avoids irrelevant syntactic complications, such as those related to the top-down and left-to right nature of the conventional concept of standardization. Actually, the extraction process was claimed to be syntactic in [Lév80] and [AsLa93]. [2]

The paper is organized as follows. In next two sections, we recall definitions of DRSs and DFSs, and some relevant standardization results for them. Section 4 gives a characterization of zig-zag relation via extraction, used in Section 5 to prove that zig-zag is a family relation in every non-duplicating stable DRS. This is the main result of this paper, and it is applied in Section 6 to define a translation of stable DCTSs into DPES. Conclusions appear in Section 7.

2 Deterministic Residual and Family Structures

In this section we recall definition of *Deterministic Residual Structures* (DRSs), which are *Abstract Reduction Systems* (ARSs) satisfying certain properties concerning residuals, and *Deterministic Family Structures* (DFSs) which are DRSs with axiomatized *family* relation on redexes with histories. The definition and some results about ARSs can be found e.g., in [Klo92]. Our definition is slightly different, and follows that of Hindley [Hin64].

Definition 2.1 An ARS is a triple $A = (Ter, Red, \rightarrow)$ where *Ter* is a set of *terms*, ranged over by t, s, o, e; *Red* is a set of *redexes* (or *redex occurrences*), ranged over by u, v, w; and $\rightarrow$: $Red \rightarrow (Ter \times Ter)$ is a function such that for any $t \in Ter$ there is only a finite set of $u \in Red$ such that $\rightarrow (u) = (t, s)$, written $t \xrightarrow{u} s$. This set will be known as the redexes of term t, where $u \in t$ denotes that u is a member of the redexes of t and $U \subseteq t$ denotes that U is a subset of the redexes. Note that $\rightarrow$ is a *total* function, so one can identify u with the triple $t \xrightarrow{u} s$. A *reduction* is a sequence $t \xrightarrow{u_1} t_2 \xrightarrow{u_2} \ldots$. Reductions are denoted by P, Q, N. We write $P : t \twoheadrightarrow s$ or $t \overset{P}{\twoheadrightarrow} s$ if P denotes a reduction (sequence) from t to s, and write $P : t \twoheadrightarrow$ if P denotes a (finite or infinite) reduction starting from t. $|P|$ denotes the length of P. $P + Q$ denotes the concatenation of P and Q. We use U, V, W to denote sets of redexes of a term.

DRSs model orthogonal term as well as graph rewrite systems, both first and higher order, and including the λ-calculus and its sharing evaluation models, with the standard Church notion of residual [Lév78, HuLé91, Klo80, Kat90, Lam90, Kha92, KKSV93, Nip93, Oos94, Raa96, Gue96]. Besides CTSs of Stark [Sta89], and ARS of Gonthier et al. [GLM92], closely related, but more syntactically oriented, models are studied in [Oos94, Mel96, Raa96].

Definition 2.2 (Deterministic Residual Structure) A *Deterministic Residual Structure* (DRS) is a pair $R = (A, /)$, where A is an ARS and $/$ is a *residual* relation on redexes relating redexes in the source and target term of every reduction $t \xrightarrow{u} s \in A$, such that for $v \in t$, the set v/u of *residuals of v under u* is a set of redexes of s; a redex in s may be a residual of only one redex in t under u, and $u/u = \emptyset$. If v has more than one u-residual, then u *duplicates* v. If $v/u = \emptyset$, then u *erases* v. A redex of s which is not a residual of any $v \in t$ under u is said to be *u-new* or *created* by u. The set of residuals of a redex under any reduction is defined by transitivity.

A *development* of a set U of redexes in a term t is a reduction $P : t \twoheadrightarrow$ that only contracts residuals of redexes from U; the development P is *complete* if U/P, the set of residuals under P of redexes from U, is empty $\emptyset$. Development of $\emptyset$ is identified with the empty reduction. U will also denote a complete development of $U \subseteq t$. The residual relation satisfies the following two axioms, called *Finite Developments (FD)* [GLM92] and *acyclicity* (which appears as axiom (4) in [Sta89]):

• [FD] All developments are terminating; all co-initial complete developments of the same set of redexes end at the same term; and residuals of a redex under all complete co-initial developments of a set of redexes are the same.

• [acyclicity] Let $u, v \in t$, let $u \neq v$, and let $u/v = \emptyset$. Then $v/u \neq \emptyset$.

We call a DRS R *non-duplicating* or *affine*, ADRS, if the residual relation in R is non-duplicating.

[2] To quote Lévy [Lév80]: 'We turn now to the hard part of this paper, which is to show that the family relation is decidable. The trouble comes from the *necessity* of looking now inside λ-expressions and from not being able to go on with algebraic manipulations'.

Non-duplicating DRSs which we study here are essentially determinate CTSs, with no distinguished start states. Having in mind possible generalization of our results to the non-affine case, we will still speak of affine DRSs, rather than DCTSs.

Similarly to [HuLé91, Lév78, Lév80, Sta89], in a DRS R the residual relation on redexes is extended to all co-initial reductions as follows: $(P_1 + P_2)/Q = P_1/Q + P_2/(Q/P_1)$ and $P/(Q_1 + Q_2) = (P/Q_1)/Q_2$, and that *Lévy-equivalence* or *permutation-equivalence* is defined as the smallest relation on co-initial reductions satisfying: $U + V/U \approx_L V + U/V$ and $Q \approx_L Q' \implies P + Q + N \approx_L P + Q' + N$, where U and V are complete developments of redex sets in the same term. Further, one defines $P \trianglelefteq Q$ iff $P/Q = \emptyset$, and can show that $P \approx_L Q$ iff $P \trianglelefteq Q$ and $Q \trianglelefteq P$; and $P \trianglelefteq Q$ iff $Q \approx_L P + N$ for some N. Below, $P \sqcup Q$ will denote $P + Q/P$. Intuitively, $P \approx_L Q$ means that P can be obtained from Q by a number of permutations of adjacent steps, therefore 'Q and P do the same work'; and $P \trianglelefteq Q$ means that P does less work than Q, the difference being Q/P, so $P + Q/P \approx_L Q$. The above relations can equivalently be defined also using Klop's method of commutative diagrams [Klo80, Bar84].

Definition 2.3 We call a DRS R *stable* (SDRS) if the following axiom is satisfied:

• [stability] If $u, v \in t$ are different redexes, $t \xrightarrow{u} e$, $t \xrightarrow{v} s$, and u creates a redex $w \in e$, then the redexes in $w/(v/u)$ are not u/v-residuals of redexes of s, i.e., they are created by u/v (see the diagram).

$$\begin{tikzcd}
t \arrow[r, "u"] \arrow[d, "v"'] & e \arrow[r, "w"] \arrow[d, twoheadrightarrow, "v/u"] & {} \\
s \arrow[r, twoheadrightarrow, "u/v"'] & o \arrow[r, twoheadrightarrow, "w/(v/u)"'] & {}
\end{tikzcd}$$

The stability axiom, and more generally Lemma 2.1 below, states that a redex cannot arise from two 'unrelated' sources. The notion of 'unrelated' is formalized by the notion of *externality*, which expresses the absence of shared (residuals of) redexes. For syntactic systems externality is a natural concept relating to overlap between components of terms involved in reduction steps.

Definition 2.4 ([GlKh96]) • Let $u \in U \subseteq t$ and $P : t \twoheadrightarrow$. We call P *external* to U (resp. u) if P does not contract residuals of redexes in U (resp. residuals of u).

• Let $P : t_0 \overset{P_i}{\twoheadrightarrow} t_i \xrightarrow{u_i} t_{i+1} \twoheadrightarrow$ and $Q : t_0 = s_0 \overset{Q_j}{\twoheadrightarrow} s_j \xrightarrow{v_j} s_{j+1} \twoheadrightarrow$. We call P *external* to Q if for any i, j, $u_i/(Q_j/P_i) \cap v_j/(P_i/Q_j) = \emptyset$ (see the diagram, where $U_{i,j} = u_i/(Q_j/P_i)$ and $V_{i,j} = v_j/(P_i/Q_j)$).

$$\begin{tikzcd}
t_0 \arrow[r, twoheadrightarrow, "P_i"] \arrow[d, twoheadrightarrow, "Q'_j"'] & t_i \arrow[r, "u_i"] \arrow[d, twoheadrightarrow, "Q_j/P_i"'] & t_{i+1} \arrow[d, twoheadrightarrow] \\
{} \arrow[r, twoheadrightarrow, "P_i/Q_j"] \arrow[d, "v_j"'] & {} \arrow[r, twoheadrightarrow, "U_{i,j}"] \arrow[d, twoheadrightarrow, "V_{i,j}"'] & {} \arrow[d, twoheadrightarrow] \\
{} \arrow[r, twoheadrightarrow] & {} \arrow[r, twoheadrightarrow] & {}
\end{tikzcd}$$

Obviously, P is external to the set U iff it is external to any development of U, and is external to a redex u iff it is external to the reduction contracting u. Note that a reduction external to one complete development of U need not be external to all developments of U, and in general, externality is not invariant under $\approx_L$. For, consider a TRS $R = \{a \to a', f(x) \to b, g(x) \to c\}$, a term $t = f(g(a))$, and reductions $P : t \xrightarrow{a} f(g(a')) \xrightarrow{f} b$, $Q : t \xrightarrow{a} f(g(a')) \xrightarrow{g} f(c)$, and $N : t \xrightarrow{g} f(c)$. Then we have $Q \approx_L N$, P is external to N, but not to Q; and P is not external to $U = \{a, g(a)\}$.

Lemma 2.1 (Stability Lemma) ([GlKh96]) Let $P : t \twoheadrightarrow s$ be external to $Q : t \twoheadrightarrow e$, in a stable DRS, and let P create redexes $W \subseteq s$. Then the residuals $W/(Q/P)$ of redexes in W are created by P/Q, and Q/P is external to W.

We now recall *Deterministic Family structures* (DFSs) which are DRSs where in addition a notion of *redex-family* is axiomatized so that the essence of sharing is captured, and all the known family notions (mentioned in the introduction) satisfy these axioms [GlKh96]. It is shown in [GlKh96] that any DFS is a stable DRS.

Definition 2.5 (Deterministic Family Structure) A *Deterministic Family Structure* (DFS for short) $\mathcal{F}$ is a triple $\mathcal{F} = (R, \simeq, \hookrightarrow)$, where R is a DRS; $\simeq$ is an equivalence relation on redexes with *histories*; and $\hookrightarrow$ is the *contribution* relation on co-initial families, defined as follows:

(1) For any co-initial reductions P and Q, a redex Qv in the final term of Q (read as v *with history* Q) is called a *copy* of a redex Pu, written $Pu \trianglelefteq_z Qv$, if $P \trianglelefteq Q$, i.e., $P + Q/P \approx_L Q$, and v is a Q/P-residual of u; the *zig-zag* relation $\simeq_z$ is the symmetric and transitive closure of the copy relation [Lév80]. The *family* relation $\simeq$ is an equivalence relation among redexes with histories containing $\simeq_z$. A *family* is an equivalence class of the family relation; families are ranged over by $\phi, \psi, \ldots$. $Fam(\ )$ denotes the family of its argument.

(2) The relations $\simeq$ and $\hookrightarrow$ satisfy the following axioms:

- [initial] Let $u, v \in t$ and $u \neq v$, in R. Then $Fam(\emptyset_t u) \neq Fam(\emptyset_t v)$.
- [contribution] $\phi \hookrightarrow \phi'$ iff for any $Pu \in \phi'$, P contracts at least one redex in ϕ.
- [creation] if $e \xrightarrow{P}\!\!\!\!\rightarrow t \xrightarrow{u} s$ and u creates $v \in s$, then $Fam(Pu) \hookrightarrow Fam((P+u)v)$.
- [termination] Any reduction that contracts redexes of a finite number of families is terminating.

3 Standardization

In this section, we recall some definitions and results from [KhGl96] concerning standardization of reductions in non-duplicating (i.e., affine) stable DRSs, ASDRSs. We define P-*needed*, P-*essential*, and P-*erased* redexes, for any reduction P, and list their (relative) properties used in this paper.

Definition 3.1 • Let $P : t \twoheadrightarrow$ and $u \in t$, in a DRS. We call u *erased* in P or P-*erased* if $u/P = \emptyset$. We say that P *discards* u if P is external to u and erases it.

• We call u P-*needed*, written $NE_P(u,t)$, if there is no $Q \approx_L P$ that is external to u, and call it P-*unneeded*, $UN_P(u,t)$, otherwise. We call u P-*essential*, $ES_P(u,t)$, if there is no $Q \approx_L P$ that discards u, and call it P-*inessential*, $IE_P(u,t)$, otherwise.

We extend these concepts to reductions co-initial with those containing u as a redex of one of its terms.

• Let $Q : t \twoheadrightarrow$, $P : t \xrightarrow{P'}\!\!\!\!\rightarrow s \twoheadrightarrow$, and $u \in s$. We say $NE_Q(u,s)$, or more precisely $NE_Q(P'u,s)$, if $NE_{Q/P'}(u,s)$. We call P Q-*needed* if so is every redex contracted in P. We call P *self-needed* if it is P-needed. The other concepts above are extended in the same way.

Note that P-neededness, P-erasure, and P-essentiality do not depend on the choice of a reduction in the class $\langle P \rangle_L$ of reductions Lévy-equivalent to P, since $u/P = u/Q$ if $P \approx_L Q$.

Lemma 3.1 Let $u \in t \xrightarrow{P}\!\!\!\!\rightarrow$, in a DRS.

(1) If u is P-erased and P-essential, then it is P-needed.

(2) If u is P-needed, then it is P-essential.

(3) If P contracts a redex v, then v is P-needed iff it is P-essential.

(4) If $P : t \twoheadrightarrow s \xrightarrow{v} o$, then v is P-needed.

Proposition 3.1 Let $P : s \twoheadrightarrow t \xrightarrow{u} e \xrightarrow{P^*}\!\!\!\!\rightarrow$, in a stable DRS.

(1) Let u create $v \in e$, and let u be P-unneeded (resp. P-inessential). Then so is v.

(2) Let $v \in e$ be a u-residual of $w \in t$, and let w be P-unneeded (resp. P-inessential). Then so is v.

(3) Let $v \in t$ be P-needed (resp. P-essential), and let $v \neq u$. Then v has at least one u-residual v', and it is the only u-residual of v, then v' is P-needed (resp. P-essential).

Definition 3.2 Let $P : t \twoheadrightarrow$ and $Q \trianglelefteq P$. The *P-needed variant of* Q, written $SE_P(Q)$, is defined as follows: let $v \in t$ be such that it is P-needed and its residual is contracted in Q first among P-needed residuals of P-needed redexes in t. Then $SE_P(Q) = v + SE_{P/v}(Q/v)$. If there is no such a redex in t, then $SE_P(Q) = \emptyset$. We call $SE_P(P)$ the *self-needed variant of* P and denote it by $SE(P)$.

Obviously, a reduction P is self-needed iff $SE(P) = P$. The notion of self-essential reduction is the best approximation to the outside-in left-to-right notion of standard reduction [Bar84, HuLé91, Klo80] for DRSs, since we do not have any nesting relation imposed on redexes, unlike ARSs of [GLM92], and there is no concept of 'left' or 'right' occurrences in DRSs. Furthermore, our concept of standardization captures the essence of the usual one in many respects. In particular, in the extraction process which we study below, self-essential reductions play the same role as outside-in left-to-right standard reductions in the extraction processes of [Lév80, AsLa93, Oos96].

Definition 3.3 We call a reduction in a DRS *standard* if it is self-essential. We write $P \approx_S Q$ if $P \approx_L Q$ and both P and Q are standard. For any standard P, we define $\langle P \rangle_S = \{Q \mid Q \approx_S P\}$.

We will use the following Standardization Theorem from [KhGl96].

Theorem 3.1 (Standardization) For any finite reduction P in a stable non-duplicating DRS, $SE(P)$ is a standard reduction Lévy-equivalent to P.

4 Equivalence of Zig-zag and Extraction

In this section, we introduce an abstract extraction algorithm and show that zig-zag related redexes (with histories) have the same canonical representatives, up to an equivalence on histories. These canonical representatives are obtained using our extraction algorithm, which leaves out all steps of histories that do not 'contribute' to the family.

Definition 4.1 Let $P : t \twoheadrightarrow s$ in an ASDRS, and let $v \in s$. We call Pv *standard* if so is P. We call Pv *canonical* if it is standard and there is no $Q \approx_L P$ such that the last step in Q does not create v.

Note that if $P \approx_S P'$, then Pv is canonical iff so is $P'v$. So canonical forms we speak of are actually objects $\langle P \rangle_S v$, for standard finite reductions P.

Lemma 4.1 Let $Q : t \stackrel{P}{\twoheadrightarrow} s \stackrel{u}{\to} e$ and let u does not create $v \in e$. Then there is a standard $Q' \approx_L Q$ such that $Q' : t \stackrel{P'}{\twoheadrightarrow} s' \stackrel{u'}{\to} e$, where $P'u' \trianglelefteq_z Pu$ (that is, $P \approx_L P' + P''$ and $u = u'/P''$) and u' does not create v.
Proof We show that $SE(Q)$ can be taken for Q'. By Definition 3.2, $SE(Q)$ is obtained from Q by a sequence of transformations $Q = Q_1, Q_2, \ldots, Q_n = SE(Q)$ such that Q_{i+1} is obtained from Q_i by permuting the first Q-needed step that has preceding Q-unneeded steps before those Q-unneeded steps (all Q_i are Lévy-equivalent). Since u is the last Q-needed step in Q by Lemma 3.1.(4), any Q_i with $i < n$ has the form $P_i + u$ such that $P_i \approx_L P$, and P_{n-1} has the form $P_{n-1} : t \stackrel{P'}{\twoheadrightarrow} o \stackrel{P''}{\twoheadrightarrow} s$ where P' is Q-needed and P'' is Q-unneeded. By Proposition 3.1.(1), P'' cannot create u, i.e., there is $u' \in o$ such that $u'/P'' = u$, and u' is Q-needed by Proposition 3.1.(2). Since P''/u' is Q-unneeded by Proposition 3.1.(2), and since the last step of $P' + u' + P''/u'$ is Q-needed by Lemma 3.1.(4), $P''/u' = \emptyset$. Since u' is Q-needed and P'' is Q-unneeded, P'' is external to u' by Proposition 3.1.(3). Hence, by the Stability Lemma, u' does not create v, and the lemma follows since $SE(Q) = P' + u'$ is standard by Theorem 3.1, and $P'u' \trianglelefteq_z Pu$ since $u = u'/P''$.

$$\begin{tikzcd}
t \arrow[r, two heads, "P'"] & o \arrow[r, two heads, "P''"] \arrow[d, "u'"'] & s \arrow[r, "u"] \arrow[d, "u"] & e \ni v \arrow[d, "\emptyset"] \\
& v \in e \arrow[r, two heads, "\emptyset"'] & e \ni v \arrow[r, two heads, "\emptyset"'] & e \ni v
\end{tikzcd}$$

Let $\langle P\rangle_S v$ not be canonical. By Definition 4.1, there is $Q : t \overset{P'}{\twoheadrightarrow} e \overset{u}{\rightarrow} s$ such that $Q \approx_L P$ and v is a u-residual of some $v' \in e$. By Lemma 4.1, Q can be chosen standard. In Q, u does not 'contribute' to v, and in the search for a shortest reduction that creates a redex in the (zig-zag) family of Qv, contraction of u can be omitted – $P'v' \simeq_z Pv$ and $|P'| < |P|$, since all standard Lévy-equivalent reductions have the same (minimal) length [KhGl96]. Obviously, reductions creating a redex in some family in a quickest way must be standard, since they are the shortest in their Lévy-equivalence classes. The transformation of Pv into $P'v$ is denoted by $Pv \overset{u}{\rightharpoonup} P'v'$, or just $Pv \rightharpoonup P'v'$; $\rightharpoonup\!\!\!\rightharpoonup$ is the transitive and reflexive closure of $\rightharpoonup$. The formal definition is as follows:

Definition 4.2 Let $Q : t \overset{P'}{\twoheadrightarrow} e \overset{u}{\rightarrow} s$ be a *standard variant* of P, i.e., a standard reduction permutation-equivalent to P, in an ASDRS, and let $v \in s$ be a u-residual of $v' \in e$. Then we write $Pv \overset{u}{\rightharpoonup} P'v'$, and call the transformation an *extraction* step. (Note that, since Q is standard, so is P' by Definition 3.1.)

Since in the above definition $|P'| < |Q| \leq |P|$, the relation $\rightharpoonup$ is trivially strongly normalizing, and in order to proof that it is confluent (modulo $\approx_S$ on histories), it is enough to prove that it is weakly confluent: $Qw'' \overset{v}{\leftharpoonup} Nw \overset{u}{\rightharpoonup} Pw'$ implies $Qw'' \overset{u'}{\rightharpoonup} N^*w^* \overset{v'}{\leftharpoonup} Pw'$. We need a lemma first whose proof can be found in the appendix.

Lemma 4.2 Let $P + u \approx_S Q + v$ and let $u \neq v$. Then there are $P'v'$ and $Q'u'$ such that $P' + v' \approx_S P$, $Q' + u' \approx_S Q$, $P' \approx_S Q'$, $P'v' \simeq_z Qv$ and $Q'u' \simeq_z Pu$, $u = u'/v'$ and $v = v'/u'$.

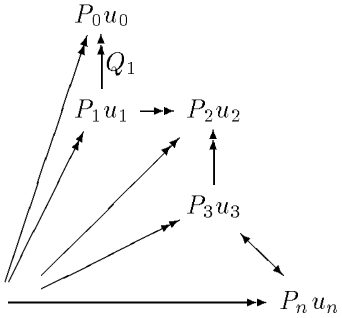
$$
\begin{tikzcd}
\cdot \arrow[r, two heads, "P'"] \arrow[d, two heads, "Q'"'] & \cdot \arrow[r, "v'"] \arrow[d, two heads, "\emptyset"] & \cdot \arrow[r, "u"] \arrow[d, two heads, "\emptyset"] & \cdot \arrow[d, two heads, "\emptyset"] \\
\cdot \arrow[r, two heads, "\emptyset"] \arrow[d, "u'"'] & \cdot \arrow[r, "v'"] \arrow[d, "u'"] & \cdot \arrow[r, "u"] \arrow[d, "u"] & \cdot \arrow[d, two heads, "\emptyset"] \\
\cdot \arrow[r, two heads, "\emptyset"'] \arrow[d, "v"'] & \cdot \arrow[r, "v"] \arrow[d, "v"'] & \cdot \arrow[r, two heads, "\emptyset"'] \arrow[d, two heads, "\emptyset"] & \cdot \arrow[d, two heads, "\emptyset"] \\
\cdot \arrow[r, two heads, "\emptyset"'] & \cdot \arrow[r, two heads, "\emptyset"'] & \cdot \arrow[r, two heads, "\emptyset"'] & \cdot
\end{tikzcd}
$$

Proposition 4.1 Every redex Pv in an ASDRS has exactly one canonical form $\langle Q'\rangle_S v'$.
Proof It is enough to show that the extraction relation $\rightharpoonup$ is weakly confluent. So let $Qw'' \overset{v}{\leftharpoonup} Nw \overset{u}{\rightharpoonup} Pw'$ with $u \neq v$ (since if $u = v$ then there is nothing to prove). We will show that $Qw'' \overset{u'}{\rightharpoonup} N^*w^* \overset{v'}{\leftharpoonup} Pw'$ for some N^*w^*, u', and v' such that $u = u'/v'$ and $v = v'/u'$. By Definition 4.2, we have from $Qw'' \overset{v}{\leftharpoonup} Nw \overset{u}{\rightharpoonup} Pw'$ that $Q + v \approx_S N' \approx_S P + u$, where N' is a standard variant of N, and $w''/v = w'/u = w$. By Lemma 4.2, we have the following situation, where $P' + v' \approx_S P$, $Q' + u' \approx_S Q$, $P' \approx_S Q'$, $u = u'/v'$, and $v = v'/u'$ (hence $P'v' \simeq_z Qv$, $Q'u' \simeq_z Pu$).

$$
\begin{tikzcd}
\cdot \arrow[r, two heads, "P'"] \arrow[d, two heads, "Q'"'] & \cdot \arrow[r, "v'"] \arrow[d, two heads, "\emptyset"] & \cdot \arrow[r, "u"] \arrow[d, two heads, "\emptyset"] & \cdot \arrow[d, two heads, "\emptyset"] \\
\cdot \arrow[r, two heads, "\emptyset"] \arrow[d, "u'"'] & \bullet \arrow[r, "v'"] \arrow[d, "u'"] & \bullet \ni w' \arrow[r, "u"] \arrow[d, "u"] & \cdot \arrow[d, two heads, "\emptyset"] \\
\cdot \arrow[r, two heads, "\emptyset"'] \arrow[d, "v"'] & \bullet \ni w'' \arrow[r, "v"] \arrow[d, "v"'] & \bullet \ni w \arrow[r, two heads, "\emptyset"'] \arrow[d, two heads, "\emptyset"] & \cdot \arrow[d, two heads, "\emptyset"] \\
\cdot \arrow[r, two heads, "\emptyset"'] & \cdot \arrow[r, two heads, "\emptyset"'] & \cdot \arrow[r, two heads, "\emptyset"'] & \cdot
\end{tikzcd}
$$

Now, it follows immediately from [stability] that there is a redex w^* in the final term of P' (and Q') such that $w^*/v' = w'$ and $w^*/u' = w''$. Thus, for $N^* = P'$, we have $Qw''\overset{u'}{-\!\!-}N^*w^*\overset{v'}{-\!\!-}Pw'$ by Definition 4.2.

Theorem 4.1 In a non-duplicating stable DRS, $Pu \simeq_z Qv$ iff they have the same unique canonical form $\langle N \rangle_S w$.
Proof By definition of $\simeq_z$, $Pu \simeq_z Qv$ implies existence of $P_0u_0 = Pu, P_1u_1, \ldots P_nu_n = Qv$ such that $P_0u_0 \trianglerighteq_z P_1u_1 \trianglelefteq_z P_2u_2 \trianglerighteq_z \ldots P_nu_n$. By the Standardization Theorem, we can take P_i to be standard.



Since $P_0u_0 \trianglerighteq_z P_1u_1$, there is Q_1 such that $P_0 \approx_L P_1 + Q_1$ and $u_0 = u_1/Q_1$. Let $P_1'u_1'$ be a canonical form of $P_1u_1 : P_1u_1 \rightharpoonup P_1'u_1'$. Then there is P_1^* such that $P_1 \approx_S P_1' + P_1^*$. We show that P_1' is $P_1' + P_1^* + Q_1$-needed, i.e., P_0-needed (since $P_1' + P_1^* + Q_1 \approx_L P_0$). Suppose on the contrary that P_1' contracts a P_0-unneeded redex. Let w be the latest P_0-unneeded step in P_1'. By Proposition 3.1.(1), w does not create the next step in P_1' (if w is not the last step in P_1'), therefore can be permuted with its next step. That w-step is again P_1'-unneeded by Proposition 3.1.(2), and can be contracted after its next step, and so on. So we can assume that w is the last step in P_1' (P_1' is chosen up to $\approx_S$). Since u_1' has a residual along $P_1^* + Q_1$, it is P_0-essential by Definition 3.1. Since w is P_0-unneeded, it is P_0-inessential by Lemma 3.1.(3). Hence w does not create u_1' by Proposition 3.1.(1). But this is impossible since $P_1'u_1'$ is canonical and w is the last step of P_1'. So we have proven that P_1' is P_0-needed. This implies that the standardization procedure of Definition 3.2 does not effect P_1' when applied to $P_1' + P_1^* + Q_1$, i.e., we can assume a standard $P_0' \approx_S P_0$ such that $P_0' = P_1' + P_0''$ for some P_0'', and $u_0 = u_1'/P_0''$. Hence $P_0u_0 \rightharpoonup P_1'u_1'$ by the definition of $\rightharpoonup$, and $P_1'u_1'$ is a canonical form of both P_0u_0 and P_1u_1. Similarly, since $P_1u_1 \trianglelefteq_z P_2u_2$, we have that $P_1'u_1'$ is a canonical form of P_2u_2, and so on. The theorem now follows from Proposition 4.1.

5 Affine Zig-zag Families

In this section we establish the main result of this paper – we show that, in ASDRSs, the zig-zag relation forms a family relation, that is, it satisfies the family axioms of DFSs. We also give a characterization of Lévy-equivalence via self-needed families.

Below, $FAM(P)$ (resp. $SFAM(P)$) denotes the set of zig-zag classes whose member (resp. P-needed, or equivalently, P-essential) redexes are contracted in P, in an ASDRS; and $Fam(Qu)$ denotes the zig-zag class of Qu. This is not in conflict with the notation in Definition 2.5, since we will show that zig-zag is a family relation.

Lemma 5.1 Let P be a standard variant of Q. Then $FAM(P) \subseteq FAM(Q)$.
Proof Let u be a redex contracted in P, say $P = P' + u + P''$. Since $P \approx_L Q$, $u/(Q/P') = \emptyset$. But since u is P-needed, it has at least one residual along Q/P' until contracted. Since Q/P' only contracts residuals of redexes contracted in Q, there is a redex v contracted in Q, say $Q = Q' + v + Q''$, such that

$u/(Q'/P') = v/(P'/Q')$, i.e., $P'u \simeq_z Q'v$, and the lemma follows.

$$\begin{tikzcd}
{} \arrow[r, two heads, "P'"] \arrow[d, "Q'"'] & {} \arrow[r, "u"] \arrow[d] & {} \arrow[r, two heads, "P''"] \arrow[d] & {} \arrow[d, "\emptyset"] \\
{} \arrow[r, two heads] \arrow[d, "v"'] & {} \arrow[r, "w"] \arrow[d, "w"] & {} \arrow[r, two heads] \arrow[d, "\emptyset"] & {} \arrow[d, "\emptyset"] \\
{} \arrow[r, two heads] \arrow[d, "Q^*"'] & {} \arrow[r, two heads, "\emptyset"'] \arrow[d] & {} \arrow[r, two heads] \arrow[d] & {} \arrow[d, "\emptyset"] \\
{} \arrow[r, two heads, "\emptyset"'] & {} \arrow[r, two heads, "\emptyset"'] & {} \arrow[r, two heads, "\emptyset"'] & {}
\end{tikzcd}$$

Lemma 5.2 If $Pu \neq Pv$, then $Fam(Pu) \neq Fam(Pv)$.
Proof Suppose on the contrary that $Fam(Pu) = Fam(Pv)$. Then they have the same canonical form by Theorem 4.1, say Qw, and we have from the extraction procedure that there is Q' such that $Q + Q' \approx_L P$ and $u = w/Q' = v$ (since w has only one residual along Q') – contradiction.

Lemma 5.3 Let P and Q be standard co-initial finite reductions. Then $P \approx_L Q$ iff $FAM(P) = FAM(Q)$.
Proof ($\Rightarrow$) Immediate from Lemma 5.1. ($\Leftarrow$) Suppose on the contrary that $P \not\approx_L Q$, and say $P/Q \neq \emptyset$. Then P contracts a redex u, say $P = P' + u + P''$, such that $u/(Q/P') \neq \emptyset$. Let v be a step in Q, i.e., $Q = Q' + v + Q''$ (see the figure for Lemma 5.1). Then if $u' = u/(Q'/P')$ and $v' = v/(P'/Q')$, we have $u' \neq v'$, thus $(P' \sqcup Q')u' \neq (P' \sqcup Q')v'$. Hence, by Lemma 5.2, $Fam(P'u) = Fam((P' \sqcup Q')u') \neq Fam((P' \sqcup Q')v') = Fam(Q'v)$, i.e., $FAM(P) \ni Fam(u) \notin FAM(Q)$ – a contradiction.

Lemma 5.4 Let $P : t_0 \twoheadrightarrow$. Then $SFAM(P) = FAM(SE(P))$.
Proof Let $P : t_0 \overset{P_i}{\twoheadrightarrow} t_i \overset{u_i}{\rightarrow} t_{i+1} \overset{P_{i+1}}{\twoheadrightarrow}$, and let $u_{i_1}, u_{i_2}, \ldots$ be all P-needed steps in P ($i_1 < i_2 < \ldots$). Then, by Definition 3.2, $SE(P) : t_0 = s_0 \overset{v_0}{\rightarrow} s_1 \overset{v_1}{\rightarrow} s_2 \twoheadrightarrow$, where u_{i_j} is the first P'_j-needed step in $P'_j = P/(v_0 + v_1 + \ldots + v_{j-1})$, and u_{i_j} is a residual of v_j along P'_j. Thus $Fam(v_j) = Fam(u_{i_j})$. And since $SFAM(P) = \{Fam(u_{i_j}) \mid j = 0, 1, \ldots\}$ and $FAM(SE(P)) = \{Fam(v_j) \mid j = 0, 1, \ldots\}$, the lemma follows.

$$\begin{tikzcd}
t_0 = s_0 \arrow[r, two heads, "P_{i_j}"] \arrow[d] & t_{i_j} \arrow[r, "u_{i_j}"] \arrow[d, "\emptyset"] & t_{i_j+1} \arrow[r, two heads, "P^*_{i_j+1}"] & P'_0 = P \\
s_j \arrow[r, two heads] \arrow[d, "v_j"'] & t_{i_j} \arrow[r, "u_{i_j}"] \arrow[d, "u_{i_j}"] & t_{i_j+1} \arrow[r, two heads, "P^*_{i_j+1}"] \arrow[d, "\emptyset"] & P'_j \\
s_{j+1} \arrow[r, two heads] & t_{i_j} \arrow[r, two heads, "\emptyset"] & t_{i_j+1} \arrow[r, two heads, "P^*_{i_j+1}"] & P'_{j+1}
\end{tikzcd}$$

Proposition 5.1 Let P and Q be co-initial finite reductions. Then $P \approx_L Q$ iff $SFAM(P) = SFAM(Q)$.
Proof $P \approx_L Q$ iff (since $P \approx_L SE(P)$ and $Q \approx_L SE(Q)$ by the Standardization Theorem) $SE(P) \approx_L SE(Q)$ iff (by the Standardization Theorem and Lemma 5.3) $FAM(SE(P)) = FAM(SE(Q))$ iff (by Lemma 5.4) $SFAM(P) = SFAM(Q)$.

Lemma 5.5 Let $Q^* : t \overset{P}{\twoheadrightarrow} s \overset{u}{\rightarrow} e$ and u create $v \in e$. Then, for any canonical form $Q'v'$ of Q^*v, Q' contracts a redex zig-zag related to Pu.
Proof We have by Lemma 4.1 that $Q = SE(Q^*) = P' + u'$, where $Q \approx_L P' + P''$ (for some Q-unneeded P'') and $u = u'/P''$. If Qv is not a canonical form, by Lemma 4.1 there is an extraction step $Qv \overset{w_1}{\rightharpoondown} Q_1 v_1$ (i.e., $Q \approx_S Q_1 + w_1$ and $v = v_1/w_1$). Since $Q_1 + w_1 \approx_S Q = P' + u'$, we have by Lemma 4.2 that $Q_1 \approx_S P_1 + u_1$ such that $P_1 u_1 \simeq_z P'u' \simeq_z Pu$. So we have $(P' + u')v \overset{w_1}{\rightharpoondown} (P_1 + u_1)v_1$ such that $P_1 u_1 \simeq_z P'u'$. Similarly, if $(P_1 + u_1)v_1$ is not a canonical form, there is an extraction step $(P_1 + u_1)v_1 \overset{w_2}{\rightharpoondown} (P_2 + u_2)v_2$ such

that $P_2u_2 \simeq_z P_1u_1 \simeq_z P'u' \simeq_z Pu$, and so on. So a canonical form of Qv has the form $(P_m + u_m)v_m$ such that $Pu \simeq_z P_m u_m$. Since, by Proposition 4.1, for any canonical form $Q'v'$ of Qv (and hence of Q^*v), $Q' \approx_S P_m + u_m$ and $v_m = v'$, it follows by Lemma 5.3 that Q' contracts a redex in the family of Pu.

Lemma 5.6 Let $Pv \stackrel{w}{\rightharpoonup} P'v'$. Then $FAM(P') \subseteq FAM(P)$.
Proof $Pv \stackrel{w}{\rightharpoonup} P'v'$ implies that $P \approx_L P' + w$, and by Lemma 5.1, $FAM(P') \subseteq FAM(P' + w) \subseteq FAM(P)$.

Definition 5.1 Let ϕ', ϕ be zig-zag classes. We write $\phi' \hookrightarrow_z \phi$ iff for any $Pu \in \phi$, P contracts a redex (with history) in ϕ'.

Lemma 5.7 Let $Q : e \stackrel{P}{\twoheadrightarrow} t \stackrel{u}{\to} s$ and let u create $v \in s$. Then $Fam(Pu) \hookrightarrow_z Fam(Qv)$.
Proof By Lemma 5.5, if $Q'v'$ is a canonical form of Qv, then $Fam(Pu) \in FAM(Q')$. Now it follows from Lemmas 5.3 and 5.6 that for any $Q^*v^* \simeq_z Qv$, $Fam(Pu) \in FAM(Q^*)$, i.e., $Fam(Pu) \hookrightarrow_z Fam(Qv)$.

Lemma 5.8 Let $P : t_0 \stackrel{P_i}{\twoheadrightarrow} t_i \stackrel{u_i}{\to} t_{i+1} \twoheadrightarrow t_n$. Then $k < m$ implies $Fam(P_k u_k) \neq Fam(P_m v_m)$.
Proof By induction on the number of zig-zag classes $\hookrightarrow_z$-contributing to $Fam(P_k u_k)$. Suppose on the contrary that $Fam(P_k u_k) = Fam(P_m v_m)$. Let $P'_k u'_k$ be a canonical form of $P_k u_k$, i.e., there is Q'_k such that $P'_k + Q'_k \approx_L P_k$, and $u_k = u'_k/Q'_k$. So we have that $P_{k+1} = P_k + u_k \approx_L P'_k + Q'_k + u_k \approx_L P'_k + u'_k + Q'_k/u'_k$. Since P'_k contracts redexes in all contributor zig-zag classes of $Fam(P'_k u'_k) = Fam(P_k u_k) = Fam(P_m u_m)$, and since by the induction assumption no redexes in these classes can be contracted again, u_m is not created by its preceding step in $u'_k + Q'_k/u'_k + u_{k+1} + \ldots + u_{m-1}$, by Lemma 5.7. Similarly, its ancestor redex is not a created redex, and so on. That is, u_m is a residual of some redex u'_m in the final term of P'_k, different from u'_k. Hence $Fam(P'_k u'_k) = Fam(P_m u_m) = Fam(P'_k u'_m)$ and $u'_k \neq u'_m$, which is not possible by Lemma 5.2 – a contradiction.

$$\begin{tikzcd}
{} \arrow[r, two heads, "P'_k"] & {} \arrow[r, two heads, "Q'_k"] \arrow[d, "u'_k"'] & {} \arrow[d, "u_k"] & & & \\
& {} \arrow[r, two heads, "Q'_k/u'_k"] & {} \arrow[r, "u_{k+1}"] & {} \arrow[r, two heads] & {} \arrow[r, "u_{m-1}"] & {}
\end{tikzcd}$$

Theorem 5.1 Let R be a non-duplicating stable DRS. Then $\mathcal{F}_\mathcal{R} = (R, \simeq_z, \hookrightarrow_z)$ is a DFS.
Proof We need to show that $\mathcal{F}_\mathcal{R}$ satisfies all family axioms. [contribution] is immediate by the definition of $\hookrightarrow_z$. Since for any $u, v \in t$, $\emptyset_t u$ and $\emptyset_t v$ are canonical forms, $u \neq v$ implies by Theorem 4.1 that $\emptyset_t u \not\simeq_z \emptyset_t v$, i.e., [initial] holds. [creation] is immediate from Lemma 5.7, and [termination] from Lemma 5.8.

6 An application

Recall that a *Prime* Event Structure (PES) [Win88] is a triple $\mathcal{E}^{\leq} = (E, Con, \leq)$, where E is a set of *events*, ranged over by $e, e_1, \ldots$; the *consistency predicate* Con is a non-empty set of subsets of E, denoted by $X, Y, \ldots$; and the *causal dependency relation* $\leq$ is a partial order on E, such that $\{e\} \in Con$, $Y \subseteq X \in Con \Rightarrow Y \in Con$, $X \in Con \wedge \exists e' \in X. e \leq e' \Rightarrow X \cup \{e\} \in Con$, and $\{e' \mid e' \leq e\}$ is finite for any $e \in E$.

We are only interested in *deterministic* structures, DPESs, where no event can prevent others from occurring, and therefore the consistency predicate is the powerset of E, and will be omitted.[3] *Configurations* (or *states*) of $\mathcal{E}^{\leq}$ are *left-closed subsets* $\alpha, \beta, \ldots$ of E, i.e., subsets $\{\alpha \subseteq E \mid e \in E \wedge e' < e \Rightarrow e' \in \alpha\}$. It is immediate from the definition of $\hookrightarrow$ that:

Theorem 6.1 ([GlKh96]) For any DFS $\mathcal{F}_t = (R_t, \simeq, \hookrightarrow)$, where R_t is a (sub)DRS whose term domain is the graph of a term t (i.e., the set of terms to which t is reducible), $\mathcal{E}^{\underline{\hookrightarrow}}_{\mathcal{F}_t} = (FAM(t), \underline{\hookrightarrow})$ is a DPES, where $\phi \underline{\hookrightarrow} \psi$ means that $\phi \hookrightarrow \psi$ or $\phi = \psi$, and $FAM(t)$ is the set of families relative to t.

[3] Determinism in ESs is defined differently in [Ren96].

Note that $\mathcal{F}_t = (R_t, \simeq, \hookrightarrow)$ holds much more information than $\mathcal{E}^{\underline{\hookrightarrow}}_{\mathcal{F}_t} = (FAM(t), \underline{\hookrightarrow})$, as in $\mathcal{E}^{\underline{\hookrightarrow}}_{\mathcal{F}_t}$ one can no longer speak of *inessentiality* of an event for a configuration of events, or speak of *equivalence* of configurations $FAM(P)$ and $FAM(Q)$ for permutation-equivalent reductions P and Q. For example, consider the DPES $\mathcal{E}^{\underline{\hookrightarrow}}_{\mathcal{F}_t}$ corresponding to the rewrite system R_t with rules $\{I(x) \to c,\ a \to b\}$ and with the graph of $t = I(a)$ as the set of terms. Its events are $I(a)\xrightarrow{a}I(b)$ and $I(a)\xrightarrow{I(a)}c$ (the steps $I(b)\xrightarrow{I(b)}c$ and $I(a)\xrightarrow{I(a)}c$ represent the same event); and its configurations are $\alpha = \{I(a)\}$, $\beta = \{a\}$, and $\gamma = \{a, I(a)\}$. Then α, β and γ are different configurations, while it is natural to identify α with γ, since the corresponding reductions are Lévy-equivalent. Thus affine DFSs provide more faithful models for orthogonal rewrite systems than DPESs, and at the same time can easily be interpreted as DPESs.

7 Conclusions and Future Work

We have shown that every non-duplicating stable DRS is in fact a DFS with the zig-zag as the family relation. The latter, when it is a comma-DFS, i.e., its term set is the graph of a single term, can be interpreted as a Deterministic Prime Event Structure (DPES) by considering families as events and the contribution relation on families as the causal dependency relation. Important examples of non-duplicating stable DRSs are several forms of graph rewriting systems, and many authors have studied the correspondence between graph rewriting systems and Prime Event Structures before [KKSV93, Sch94, CELMR94, CK95]. Our translation is different, and seems conceptually more clear as it is for Abstract Rewrite Systems. And we construct an affine DFS model (which is more powerful and informative than the DPES model) *directly*, unlike the above approaches [Sch94, CELMR94] which construct a DPES model by constructing a corresponding trace language first, and applying known results relating Mazurkiewicz traces with Event Structures [Bed87]. Our translation is close in spirit to that in [KKSV93], but the latter construction uses syntax of orthogonal Term Graph Rewriting Systems heavily, and our proofs are completely different.

There are three immediate directions to continue this work. First, the possibility of constructing DFSs from duplicating stable DRSs must be investigated. In the general case, the [termination] axiom of DFSs becomes a strong requirement, and extra axioms on stable DRSs may be needed to ensure it. We expect that a similar extraction algorithm will be applicable, but proofs will become more complicated. Event Structure semantics for orthogonal rewrite systems with duplicating residual relation are studied, among others, in [LaMo92, Lan93], but the results there are limited because of the problems with erasure illustrated by the example in Section 6.

Second, it is natural to extend DPESs with an axiomatized *erasure* relation or an axiomatized *permutation-equivalence* relation to enable DPESs to give an adequate semantics to orthogonal rewrite systems, and to cope with the example in the previous section. This is indeed possible, and in an forthcoming paper we show that such refined event models are isomorphic to affine DFSs, therefore can serve as faithful event models for orthogonal rewrite systems. (This approach, in particular, solves the problems with erasure discussed in [Lan93], Chapter 8, and in [KKSV93]).

Finally, we mention that non-deterministic Residual Structures must be considered as well. For example, normalization by neededness theory for non-orthogonal systems is studied in Boudol [Bou85, Mel96], and constructions of Event Structures from non-orthogonal graph rewriting systems are studied, among others, in [CELMR94, CK95].

References

[AsLa93] Asperti A., Laneve C. Interaction Systems I: the theory of optimal reductions. MSCS, 11:1-48, Cambridge University Press, 1993.

[AsLa96] Asperti A., Laneve C. Interaction Systems II: the practice of optimal reductions. TCS, 159(2):191-244, 1996.

[Bar84] Barendregt H. P. The Lambda Calculus, its Syntax and Semantics. North-Holland, 1984.

[BBKV76] Barendregt H.P., Bergstra J., Klop J.W., Volken H. Degrees, reductions and representability in the λ-calculus. Report 22, University of Utrecht, 1976.

[Bed87] Bednarczyk M. A. Categories of asynchronous systems. PhD Thesis, University of Sussex, Brighton, 1987.

[Ber79] Berry G. Modéles complétement adéquats et stables des λ-calculs typés. Thèse de l'Université de Paris VII, 1979.

[Bou85] Boudol G. Computational semantics of term rewriting systems. In: Algebraic methods in semantics. Nivat M., Reynolds J.C., eds. Cambridge Univ. Press, 1985, p. 169-236.

[CK95] Clark D., Kennaway R. Some properties of non-orthogonal term graph rewriting. Electronic Notes in Theoretical Computer Science, vol. 2, A. Corradini, U. Montanari, eds. Elsevier, 1995, p. 141-150. To appear in MSCS.

[CELMR94] Corradini A., Ehrig H., Löwe M., Montanari U., Rossi F. An event structure semantics for safe graph grammars. In proc PROCOMET'94, IFIP Transactions A-56, 1994.

[GlKh96] Glauert J.R.W., Khasidashvili Z. Relative normalization in deterministic residual structures. In: Proc. of the 19^{th} International Colloquium on Trees in Algebra and Programming, CAAP'96, Springer LNCS, vol. 1059, H. Kirchner, ed. 1996, p. 180-195.

[GAL92] Gonthier G., Abadi M., Lévy J.-J. The geometry of optimal lambda reduction. In: POPL'92, p.15-26.

[GLM92] Gonthier G., Lévy J.-J., Melliès P.-A. An abstract Standardisation theorem. In: LICS'92, p. 72-81.

[Gue96] Guerrini S. Theoretical and practical issues of optimal implementations of functional languages. Ph.D. Thesis, University of Pisa, 1996.

[Hin64] Hindley R.J. An abstract form of the Church-Rosser theorem I. JSL, 34(4):545-560, 1969.

[HuLé91] Huet G., Lévy J.-J. Computations in Orthogonal Rewriting Systems. In: Computational Logic, Essays in Honor of Alan Robinson, J.-L. Lassez and G. Plotkin, eds. MIT Press, 1991.

[HP91] Hoffmann B., Plump D. Implementing term rewriting by Jungle evaluation. Theoretical Informatics and Applications 25:445-472, 1991.

[Kat90] Kathail V. Optimal Interpreters for Lambda-calculus based functional languages, PhD thesis, MIT, 1990.

[KeSl89] Kennaway J. R., Sleep M. R. Neededness is hypernormalizing in regular combinatory reduction systems. Report. University of East Anglia, 1989.

[KKSV93] Kennaway J. R., Klop J. W., Sleep M. R, de Vries F.-J. Event structures and orthogonal term graph rewriting. In: M. R. Sleep, M. J. Plasmeijer, and M. C. J. D. van Eekelen, eds. Term Graph Rewriting: Theory and Practice. John Wiley, 1993.

[Kha92] Khasidashvili Z. The Church-Rosser theorem in Orthogonal Combinatory Reduction Systems. Report 1825, INRIA Rocquencourt, 1992.

[KhGl96] Khasidashvili Z., Glauert J. R. W. Discrete normalization in Deterministic Residual Structures. In proc. of the 5^{th} International Conference on Algebraic and Logic Programming, ALP'96, Springer LNCS, vol. 1139, M. Hanus, M. Rodríguez-Artalejo, eds. 1996, p.135-149.

[Klo80] Klop J. W. Combinatory Reduction Systems. Mathematical Centre Tracts n. 127, CWI, Amsterdam, 1980.

[Klo92] Klop J. W. Term Rewriting Systems. In: S. Abramsky, D. Gabbay, and T. Maibaum eds. Handbook of Logic in Computer Science, vol. II, Oxford University Press, 1992, p. 1-116.

[Lam90] Lamping J. An algorithm for optimal lambda calculus reduction. In 17^{th} POPL, 1990, p.16-30.

[Lan93] Laneve C. Optimality and Concurrency in Interaction Systems. Ph.D. Thesis, University of Pisa, 1993.

[LaMo92] Laneve C., Montanari U. Axiomatizing permutation equivalence in the λ-calculus. In proc. of the 3^{rd} International Conference on Algebraic and Logic Programming, ALP'92, Springer LNCS, vol. 632, 1996, p.350-363.

[Lév78] Lévy J.-J. Réductions correctes et optimales dans le lambda-calcul, Thèse de l'Université de Paris VII, 1978.

[Lév80] Lévy J.-J. Optimal reductions in the Lambda-calculus. In: To H. B. Curry: Essays on Combinatory Logic, Lambda-calculus and Formalism, Hindley J. R., Seldin J. P. eds, Academic Press, 1980, p. 159-192.

[Mar91] Maranget L. Optimal derivations in weak λ-calculi and in orthogonal Term Rewriting Systems. In: Proc. POPL'91, p. 255-269.

[NPW81] Nielsen M., Plotkin G., Winskel G. Petri nets, event structures and Domains. Part 1. TCS 13:85-108, 1981.

[Nip93] Nipkow T. Orthogonal higher-order rewrite systems are confluent. In: Proc. of the 1^{st} International Conference on Typed Lambda Calculus and Applications, TLCA'93, Springer LNCS, vol. 664, Bazem M., Groote J.F., eds. Utrecht, 1993, p. 306-317.

[Mel96] Melliès P.-A. Description Abstraite des Systèmes de Réécriture. Thèse de l'Université Paris 7, 1996.

[Oos94] Van Oostrom V. Confluence for Abstract and Higher-Order Rewriting. Ph.D. Thesis, Free University of Amsterdam, 1994.

[Oos96] Van Oostrom V. Higher order families. In proc. of RTA'96, Springer LNCS, vol. 1103, Ganzinger, H., ed., 1996, p. 392–407.

[Raa96] Van Raamsdonk F. Confluence and normalisation for higher-order rewriting. Ph.D. Thesis, CWI Amsterdam, 1996.

[Ren96] Rensink A. Denotational, Causal, and Operational Determinism in Event Structures. In: Proc. of the 19^{th} International Colloquium on Trees in Algebra and Programming, CAAP'96, Springer LNCS, vol. 1059, H. Kirchner, ed. 1996, p. 272-286.

[Sch94] Schied G. On relating rewrite systems and graph grammars to event structures. In [SE94], p. 326-340.

[SE94] Schneider H.-J. Ehrig H., editors. Proceedings of the Dagstuhl Seminar 9301 on Graph Transformations in Computer Science, Springer LNCS, vol. 776, 1994.

[Sta89] Stark E. W. Concurrent transition systems. Theoretical Computer Science, vol. 64, 1989, p. 221-270.

[Wad71] Wadsworth C.P. Semantics and pragmatics of the Lambda-calculus. PhD. thesis, University of Oxford, 1971.

[Win88] Winskel G. Event Structures. Univ. of Cambridge, Computer Laboratory, Report no. 95, 1988.

8 Appendix: Proof of Lemma 4.2

We use the following two lemmas from [GlKh96], and three simple new lemmas in this proof.

Lemma 8.1 ([GlKh96]) Let $P : t \twoheadrightarrow s$ be external to a complete development N of $F \subseteq t$, in a DRS, and let $Q : t \twoheadrightarrow o$, then P/Q is external to N/Q.

Lemma 8.2 ([KhGl96]) Let $u \in t_0 \overset{P}{\twoheadrightarrow} t_k$, and let P be external to u, in a DRS. Then $u \not\approx_L P$.

Lemma 8.3 Let $u + P \approx_S Q + u'$, where $u' = u/Q$ and $P = v + P'$. Then u does not create v, and u can be contracted after v, i.e., $u + v \approx_L v^* + u^*$, where $v = v^*/u$ and $u^* = u/v^*$. Further, $v^*/Q = \emptyset$ and $u' = u^*/(Q/v^*)$.

Proof Let $Q' = Q/u$. Since $u + P$ is standard, so is P by Definition 3.3 and Definition 3.1, so v is P-needed, and since $P \approx_L Q'$, v is Q-needed too, i.e, Q' contracts a residual v' of v. Since Q' contracts residuals of redexes contracted in Q, Q contracts a redex v'' whose residual is v'. So we have the following picture:

$$\begin{tikzcd}
{} \arrow[r, "u"] \arrow[d] & {} \arrow[r, "v"] \arrow[d] & {} \arrow[r, twoheadrightarrow, "P'"] \arrow[d] & {} \arrow[d] \\
{} \arrow[r] \arrow[d, "v''"'] & {} \arrow[r, "v'"] \arrow[d, "v'"'] & {} \arrow[r, twoheadrightarrow] \arrow[d] & {} \arrow[d] \\
{} \arrow[r, twoheadrightarrow] \arrow[d] & {} \arrow[r, twoheadrightarrow] \arrow[d] & {} \arrow[r, twoheadrightarrow] \arrow[d] & {} \arrow[d] \\
Q \arrow[r, "u'"'] & Q' \arrow[r, twoheadrightarrow, "\emptyset"'] & {} \arrow[r, twoheadrightarrow] & {}
\end{tikzcd}$$

Now, since Q is external to u (since u has a Q-residual u' and the DRS is non-duplicating), we have immediately by the Stability Lemma that both v and v'' are residuals of some redex v^* in the initial term. Hence $u + v \approx_L v^* + u^*$, where $v = v^*/u$ and $u^* = u/v^*$. Since Q contracts a residual v'' of v^*, $v^*/Q = \emptyset$. Since Q is external to u, we have by Lemma 8.1 that $Q^* = Q/v^*$ is external to u^*. Since u is $u + P$-needed, so is u^* by Proposition 3.1.(3). Hence u^* is $Q^* + u'$-needed. Since Q^* is external to u^* and $Q^* + u'$ contracts a residual of u^*, we have $u' = u^*/Q^*$.

$$\begin{tikzcd}
\bullet \arrow[r, "v^*"] \arrow[d, "Q"'] & \bullet \arrow[r, "u^*"] \arrow[d, "Q^*"'] & \bullet \arrow[r, two heads, "P'"] \arrow[d] & \bullet \arrow[d] \\
\bullet \arrow[r, two heads, "\emptyset"] \arrow[d, "u'"'] & \bullet \arrow[r, "u'"] \arrow[d, "u'"'] & \bullet \arrow[r, two heads] \arrow[d, "\emptyset"] & \bullet \arrow[d] \\
\bullet \arrow[r, two heads, "\emptyset"'] & \bullet \arrow[r, two heads, "\emptyset"'] & \bullet \arrow[r, two heads] & \bullet
\end{tikzcd}$$

Lemma 8.4 Let $P + u \approx_S Q + v$ and let $u \neq v$. Then $P \not\approx_L Q$.
Proof Suppose on the contrary that $P \approx_L Q$. Then $P + u \approx_S Q + v$ iff $u \approx_L v$. But, by [acyclicity], this is only possible when $u = v$ – a contradiction.

Lemma 8.5 Let $P \approx_S Q$. Then any non-empty step in Klop's diagram of P and Q in P-needed.
Proof Since every step in the diagram is a residual of a redex contracted in P or Q, the lemma follows immediately from Proposition 3.1.(3).

Proof of Lemma 4.2 Since $P+u \approx_S Q+v$, we have $v \approx_L (P+u)/Q$. By Lemma 8.2, $(P+u)/Q$ contracts a residual of v. We show that $P/Q \neq \emptyset$.

$$\begin{tikzcd}
\bullet \arrow[r, two heads, "P"] \arrow[d, "Q"'] & \bullet \arrow[r, "u"] \arrow[d, "Q/P"', "u"] & \bullet \arrow[d, "\emptyset"] \\
\bullet \arrow[r, two heads, "P/Q", "v"'] \arrow[d, "v"'] & \bullet \arrow[r, two heads] \arrow[d] & \bullet \arrow[d, "\emptyset"] \\
\bullet \arrow[r, two heads, "\emptyset"'] & \bullet \arrow[r, two heads, "\emptyset"'] & \bullet
\end{tikzcd}$$

Suppose on the contrary that $P/Q = \emptyset$. Then, by [acyclicity], $v = u/(Q/P)$. Further, by Lemma 8.4, $P \not\approx_L Q$, hence $P/Q = \emptyset$ implies $Q/P \neq \emptyset$. But $P + u \approx_L Q + v$ implies $(Q/P)/u = \emptyset$. Since $Q + v$ is standard, the first (and any other) step of Q whose residual, say w, is contracted in Q/P is u-needed by Lemma 8.5. Hence $w/u = \emptyset$ implies $u = w$, and therefore $Q/P = u$ and $u/(Q/P) = \emptyset$, contrary to $v = u/(Q/P)$. So $P/Q \neq \emptyset$. Since P is $P+u$-needed (recall that $P+u$ is standard), so is P/Q, i.e., P/Q is v-needed. Hence, by [acyclicity], the first (and the only) step of P/Q coincides with v, i..e., $P/Q = v$. Thus P contracts a redex v'' whose residual is v. So if $P = P_1 + v'' + P_2$, then $(Q + v)/P_1 = Q/P_1 + v$, and we have $v'' + P_2 + u \approx_S Q/P_1 + v$ and $v'' + P_2 \not\approx_L Q/P_1$ (see the figure).

$$\begin{tikzcd}
\bullet \arrow[r, two heads, "P_1"] \arrow[d, "Q"'] & \bullet \arrow[r, "v''"] \arrow[d] & \bullet \arrow[r, two heads, "P_2"] \arrow[d] & \bullet \arrow[r, "u"] \arrow[d] & \bullet \arrow[d, "\emptyset"] \\
\bullet \arrow[r, two heads, "\emptyset"] \arrow[d, "v"'] & \bullet \arrow[r, "v"] \arrow[d, "v"'] & \bullet \arrow[r, two heads, "\emptyset"] \arrow[d, "\emptyset"] & \bullet \arrow[r, two heads, "\emptyset"] \arrow[d, "\emptyset"] & \bullet \arrow[d, "\emptyset"] \\
\bullet \arrow[r, two heads, "\emptyset"'] & \bullet \arrow[r, two heads, "\emptyset"'] & \bullet \arrow[r, two heads, "\emptyset"'] & \bullet \arrow[r, two heads, "\emptyset"'] & \bullet
\end{tikzcd}$$

Now, by repeated application of Lemma 8.3, $v'' + P_2 + u$ can be transformed into a reduction $P_2' + v' + u$ such that $v'' + P_2 \approx_S P_2' + v'$, $v' = v''/P_2'$, and $v = v'/(Q/(P_1 + P_2'))$.

$$\begin{tikzcd}
{} \arrow[r, "P_1"] \arrow[d, "Q"'] & {} \arrow[r, two heads, "P_2'"] \arrow[d] & {} \arrow[r, "v'"] \arrow[d] & {} \arrow[r, "u"] \arrow[d] & {} \arrow[d, "\emptyset"] \\
{} \arrow[r, two heads, "\emptyset"] \arrow[d, "v"'] & {} \arrow[r, two heads, "\emptyset"] \arrow[d, "v"'] & {} \arrow[r, "v"] \arrow[d, "v"'] & {} \arrow[r, two heads, "\emptyset"] \arrow[d] & {} \arrow[d, "\emptyset"] \\
{} \arrow[r, two heads, "\emptyset"'] & {} \arrow[r, two heads, "\emptyset"'] & {} \arrow[r, two heads, "\emptyset"'] & {} \arrow[r, two heads, "\emptyset"'] & {}
\end{tikzcd}$$

Hence, if we take $P' = P_1 + P_2'$, we have that $P' + v' \approx_S P$ and $P'v' \trianglelefteq_z Qv$. Existence of $Q'u'$ such that $Q' + u' \approx_S Q$ and $Q'u' \trianglelefteq_z Pu$ can be shown similarly. Since $P_2/(Q/(P_1 + v'')) = \emptyset$, we have again by repeated application of Lemma 8.3 that $P'/Q \approx_L P'/(Q' + u') \approx_L (P'/Q')/u' = \emptyset$. But P'/Q' is external to u' since u' has a $P/Q' \approx_L P'/Q' + v'/(Q'/P')$-residual (by $Q'u' \trianglelefteq_z Pu$).

$$\begin{tikzcd}
{} \arrow[r, two heads, "P'"] \arrow[d, "Q'"'] & {} \arrow[r, "v'"] \arrow[d, "\emptyset"] & {} \arrow[r, "u"] \arrow[d, "\emptyset"] & {} \arrow[d, "\emptyset"] \\
{} \arrow[r, two heads, "\emptyset"] \arrow[d, "u'"'] & {} \arrow[r, "v'"] \arrow[d, "u'"] & {} \arrow[r, two heads, "u"] \arrow[d, "u"] & {} \arrow[d, "\emptyset"] \\
{} \arrow[r, two heads, "\emptyset"'] \arrow[d, "v"'] & {} \arrow[r, "v"] \arrow[d, "v"'] & {} \arrow[r, two heads, "\emptyset"'] \arrow[d, "\emptyset"] & {} \arrow[d, "\emptyset"] \\
{} \arrow[r, two heads, "\emptyset"'] & {} \arrow[r, two heads, "\emptyset"'] & {} \arrow[r, two heads, "\emptyset"'] & {}
\end{tikzcd}$$

Hence we have by Lemma 8.2 and Lemma 8.5 that $P'/Q' = \emptyset$. The converse is proved similarly, so $P' \approx_L Q'$. It follows that $u = u'/v'$ and $v = v'/u'$.

This article was processed using the LaTeX macro package with LLNCS style