Relating Conflict-free Stable Transition and Event Models (Extended Abstract, Submitted to LICS'97)

Zurab Khasidashvili
NTT Basic Research Laboratories, Room A4S-220S
Atsugi-shi, Kanagawa-ken, 243-01, Japan
zurab@theory.brl.ntt.jp *

John Glauert
School of Information Systems, UEA
Norwich NR4 7TJ England
jrwg@sys.uea.ac.uk †

Abstract

We describe an event structure style semantics for conflict-free (term and graph, possibly with bound variables) rewrite systems, including the λ-calculus, and other stable transition systems with residual relation. Our interpretation is based on interpreting redex-families as events. It treats permutation-equivalent reductions as representing the same concurrent computation. Since Event Structures are inadequate for such interpretation (because of the erasure of redexes), we extend the Prime Event Structure model by axiomatizing the permutation-equivalence on finite configurations in two different ways (for the conflict-free case), and show that thus extended models are equivalent to known stable transition models with axiomatized residual and family relations.

1 Introduction

The prime goal of this paper is to provide a fully adequate concurrent semantics for orthogonal rewrite systems. As an abstract model of concurrency, we chose Event Structures, the most successful and commonly accepted one [NPW81, Win86/89]. Conflict-free Prime Event Structures (PESs) are sufficient for our purposes. A (conflict-free) PES is simply a set of events ordered by *causal dependency* relation where an event can only dominate a finite number of events. The idea is to view a process as performing events (some *atomic* tasks), and an event can only occur if all events on which it depends already have. Causally independent events are *concurrent* events and can be evaluated in parallel. Stages of a computation are thus represented by *configurations* – left-closed event sets w.r.t. the causal order, and the same configuration represents all sequential computations executing events in that configuration, ignoring the order in which the concurrent events have occurred.

In the theory of orthogonal rewrite systems (such as the λ-calculus) there is the well established concept of 'redex-family', due to Lévy [Lév80], which formalizes the idea of the 'same atomic task'. Therefore we want to base our interpretation on interpreting redex-families as events, hence to interpret family-reductions (contracting entire families as multi-steps) as configurations of the corresponding PES. This is also justified by the fact that reductions in optimal graph-implementations of orthogonal systems precisely correspond to family-reductions in the original system.

Recent advances in abstract study of orthogonal rewrite systems [Sta89, GLM92, GK96, KG96, KG96a] allows us to address the problem in an abstract setting. Stable Deterministic Residual Structures (SDRSs) and Deterministic Family Structures (DFSs) model computation in orthogonal rewrite systems. DRSs are Abstract Rewrite Systems with an axiomatized residual relation; DFSs are DRSs there in addition the concept of redex-family is axiomatized. Stable DRSs allow [GK96] to prove analogs of the Normalization and Standardization Theorems [Bar84]. In DFSs one can further prove Lévy's Optimality Theorem and the Unique Families Lemma. The latter states that any family can be contracted at most once in a family-reduction. It corresponds to the fact that any event may occur at most once in the course of a computation, and is needed for event structure interpretation of DRSs based on interpreting families as events.

Since we want to restrict our attention to family-reductions only, we define *linearization* of any DFS, which is a non-duplicating, also called affine, SDRS,

*The contact author. Tel: 81 462 403674, Fax: 81 462 404709, Email from January: zurab@theory.brl.ntt.co.jp.

†Part of this work was supported by the Engineering and Physical Sciences Research Council of Great Britain under grant GR/H 41300.

ASDRS. We have already shown [KG96a] that any ASDRS forms a DFS with the *zig-zag* [Lév80] as the family relation. Family-reductions in a DFS precisely correspond to reductions in its linearization, and this allows us to concentrate on construction of event semantics for affine SDRSs only.

To view a process as the domain of all its configurations, an *adequacy* requirement is desirable. Namely, *runs* of a (finite) configuration (i.e., all its possible sequentializations) should be in one-to-one correspondence with all reductions in the equivalence class which the configuration is supposed to represent. Because of the huge importance of the *permutation* or *Lévy* equivalence [Lév80] to the whole theory of (not only orthogonal) rewrite systems [Bou85], which is well understood by now, and to the concurrency theory too [Sta89], we want to treat permutation-equivalent reductions as representing the same concurrent computation, as indeed permutation-equivalent reductions result one from another by permuting concurrent consecutive steps. However, this is clearly impossible in the present PES framework [Win86/89], as a configuration in a PES has only a finite number of runs while a permutation-equivalence class of a (finite) reduction may have an infinite number of elements.

In order for an equivalence relation to satisfy the adequacy requirement, the corresponding equivalence classes of (finite) reductions must be partially ordered in a way to form distributive lattices, as the domains of (finite) configurations of a PES ordered by inclusion are such lattices [Win86/89]. However, it is well known that Lévy's reduction space is not in general a lattice, but only an upper semi-lattice, as the greatest lower bound of two co-initial reductions need not exist [Lév80]. This problem, caused by erasure of redexes, is well discussed in [Lan93], where it is argued that event semantics say for the λ-calculus cannot therefore be based on permutation-equivalence.

Several authors have proposed different ways to get round of this problem. Laneve [Lan93] introduces the *distributive* equivalence on β-reductions which only allow permutation of steps that cannot erase or duplicate one another, and consequently all equivalent sequences have the same length. Kennaway et al [KKSV93] do work with permutation equivalence, for orthogonal Term Graph Rewriting, but they only restrict themselves to needed events of a normalizable term to cope with problems with erasure. As needed redexes cannot be erased, all needed normalizing reductions are of the same length, and on needed reductions, the permutation and distributive equivalences coincide (since the system is affine). Similarly, Clark and Kennaway [CK95] work with permutation-equivalence and allow infinite reductions but restrict themselves to standard reductions only (which are unique in their permutation-equivalence class). Schied [Sch94] and Corradini et al [CELMR94] work with linear rewrite systems (no duplication, no erasure), mainly graph grammars, and do not therefore encounter the erasure problems.

In order to cope with the adequacy problem, we extend the PES model by axiomatizing permutation-equivalence on event configurations. We do it in two different ways. One by directly constraining the class of all equivalence relations on configurations, and another by axiomatizing the *inessentiality* or *erasure* relation expressing redundancy of an event for a configuration. The latter extends by further axioms an incomplete axiomatization of the inessentiality relation in [GK96]. In the resulting extended models, Deterministic *Permutation* and *Erasure* Event Structures, DPESs and DEESs, we can identify configurations with different number of events but still representing the same permutation-equivalence class of reductions. We show that permutation-equivalence classes of configurations in a DPES/DEES, appropriately ordered, are isomorphic to Lévy's reduction space of the corresponding ASDRSs, therefore the former give a *fully adequate* semantics to ASDRSs.

The paper is organized as follows. In the next section, we review the existing theory of DRSs and DFSs used in this paper. In section 3 we introduce DPESs and DEESs and establish their equivalence. In section 4 we relate the latter event models with affine SDRSs and zig-zag DFSs, showing the equivalence of the four models. In section 5, we define Lévy-embedding on configurations in a DPES/DEES and show its isomorphism with Lévy's reduction space of the corresponding ASDRS. Finally, in section 6, we extend our interpretation to duplicating orthogonal rewrite systems by encoding the latter into ASDRSs using the linearization method mentioned above. Conclusions appear in section 7.

2 Deterministic Residual and Family Structures

In this section we recall some basic theory for DRSs and DFSs developed in [GK96, KG96, KG96a].

Definition 2.1 ([Hin64]) 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 $\rightarrow : Red \mapsto (Ter \times$

Ter) is a function such that for any $t \in Ter$ there is only a finite set of $u \in Red$ such that $\mapsto(u) = (t, s)$, 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. $P + Q$ denotes the concatenation of P and Q. u also denotes the reduction that contracts u.

DRSs are similar to Stark's *Determinate Concurrent Transition Systems* (DCTSs) [Sta89] and *Abstract Reduction Systems* of Gonthier et al. [GLM92]. Unlike DCTSs, the residual relation in DRSs may be duplicating, and unlike ARSs of [GLM92], we do not have a nesting relation on redexes. Several refined concepts of abstract rewriting are studied in [Oos94, Mel96, Raa96]. DRS are intended to model all orthogonal term and graph rewrite systems, both first and higher-order, and other conflict-free transition systems with a reasonable level of abstraction, while say ARSs of [GLM92] do not cover e.g., orthogonal term graph rewriting systems [SPE93].

Definition 2.2 A *Deterministic Residual Structure* (DRS) is a pair $\mathcal{R} = (A, /)$, 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* (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 $\mathcal{R}$ *stable* (SDRS) if the following axiom is satisfied:

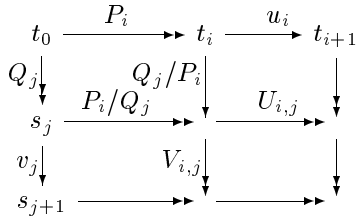
- [stability] If $u, v \in t$ are different redexes, $t \xrightarrow{u} e$, $t \xrightarrow{v} s$, and u creates a redex $w \in e$, then the redexes in $w/(v/u)$ are not u/v-residuals of redexes of s, i.e., they are created by u/v .

In a DRS $\mathcal{R}$, the residual relation on redexes is extended to all co-initial reductions exactly as in syntactic orthogonal rewrite systems [HL91, Lév78, Lév80, Sta89]: $(P_1 + P_2)/Q = P_1/Q + P_2/(Q/P_1)$ and $P/(Q_1 + Q_2) = (P/Q_1)/Q_2$, and *Lévy-equivalence* or *permutation-equivalence* is defined as the smallest relation on co-initial reductions satisfying: $U + V/U \approx V + U/V$ and $Q \approx Q' \Rightarrow P + Q + N \approx P + Q' + N$, 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 $P \approx Q$ iff $P \trianglelefteq Q$ and $Q \trianglelefteq P$; and $P \trianglelefteq Q$ iff $Q \approx P + N$ for some N. Finally, one shows that for any co-initial reductions P, Q, $P \sqcup Q \approx Q \sqcup P$, where $P \sqcup Q = P + Q/P$, and is the greatest lower bound of P and Q in the reduction space ordered by Lévy's embedding relation $\trianglelefteq$.

The following two lemmas, established in [GK96, KG96], are fundamental in proving properties in SDRSs. They extend [acyclicity] and [stability] axioms from one step reductions to any co-initial *external* reductions, that is, reductions that do not contact redexes having common residuals.

Definition 2.3 • Let $u \in U \subseteq t$ and $P : t \twoheadrightarrow o$. We call P *external* to U (resp. u) if P does not contract residuals of redexes in U (resp. residuals of u).

• Let $P : t_0 \overset{P_i}{\twoheadrightarrow} t_i \xrightarrow{u_i} t_{i+1} \twoheadrightarrow t_n$ and $Q : t_0 = s_0 \overset{Q_j}{\twoheadrightarrow} s_j \xrightarrow{v_j} s_{j+1} \twoheadrightarrow s_m$. 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 $V_{i,j} = v_j/(P_i/Q_j)$).



Lemma 2.4 (Stability Lemma) 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.

Lemma 2.5 (Acyclicity Lemma) Let P, N be co-initial finite reductions in a DRS, and let P be external to N. Then $N \not\approx P$.

Definition 2.6 A *Deterministic Family Structure* (DFS) is a triple $\mathcal{F} = (\mathcal{R}, \simeq, \hookrightarrow)$, where $\mathcal{R}$ 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 if $P \trianglelefteq Q$, i.e., $P+Q/P \approx Q$, 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. 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) Further, $\simeq$ and $\hookrightarrow$ satisfy the following axioms:

• [initial] Let $u, v \in t$ and $u \neq v$, in $\mathcal{R}$. Then $Fam(\emptyset_t u) \neq Fam(\emptyset_t v)$, where $\emptyset_t$ is the empty reduction starting from t.

• [contribution] $\phi \hookrightarrow \phi'$ iff for any $Pu \in \phi'$, P contracts at least one redex in ϕ.

• [creation] if $e \xrightarrow{P} t \xrightarrow{u} s$ 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.

We call a DFS $\mathcal{F}$ a *zig-zag* DFS, ZDFS, if its family relation is the zig-zag $\simeq_z$ (the corresponding contribution relation $\hookrightarrow_z$ is determined by [contribution]).

It is shown in [GK96] that every DFS is a stable DRS.

Convention In this paper we only consider *comma-DRS*s and *comma-DFS*s. That is, the term set of any DRS will be the *reduction graph* of a term, called the *initial* term, i.e., the set of terms to which the initial term is reducible. The initial term will often be denoted by $\emptyset$, by analogy with the initial state in an Event Structure, although $\emptyset$ will also be used to denote empty reductions. Further, in DFSs, families will always be considered relative to $\emptyset$, i.e., all histories start with $\emptyset$. Unless otherwise stated, by a *reduction* we mean a *finite* reduction. Reductions starting from $\emptyset$ will also be called *initial* reductions.

2.1 Standardization

We recall some definitions and results from [KG96] concerning standardization of reductions in non-duplicating, i.e., affine, SDRSs, ASDRSs.

Definition 2.7 • Let $P : t \twoheadrightarrow o$ 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* if there is no $Q \approx P$ that is external to u, and call it P-*unneeded* otherwise. We call u P-*essential* if there is no $Q \approx P$ that discards u, and P-*inessential* otherwise.

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

• Let $Q : t \twoheadrightarrow o$, $P : t \xrightarrow{P'} s \twoheadrightarrow e$, and $u \in s$. We say that u is Q-*needed*, or more precisely, $P'u$ is Q-*needed*, if u is Q/P'-needed. 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 $u/P = u/Q$ when $P \approx Q$. The *discards* concept however does depend on the particular reduction in the permutation-equivalence class.

Lemma 2.8 Let $P : s \twoheadrightarrow t \xrightarrow{u} e \twoheadrightarrow o$ in an ASDRS.

(1) $w \in t$ is P-needed iff it is P-erased and P-essential.

(2) If $P : t \twoheadrightarrow s' \xrightarrow{w} o$, then $w \in s'$ is P-needed.

(3) If u creates $v \in e$ and u is P-unneeded (resp. P-inessential), then so is v.

(4) If $u \neq v \in t$, then v is P-needed (P-essential) iff v has a P-needed (P-essential) residual in e.

Self-needed reductions play the role of *standard* reductions in SDRSs, and the following algorithm is a standardization procedure for reductions in ASDRSs. Let $P : t \twoheadrightarrow s$. The *canonical standard variant of* P, $ST(P)$, is defined as follows: If $P = \emptyset$, when so is $ST(P)$. Otherwise, let $v \in t$ be such that it is P-needed and its residual is contracted in P first among P-needed residuals of P-needed redexes in t (existence of such v follows from Lemma 2.8). Then $ST(P) = v + ST(P/v)$. We write $Q \in STV(P)$ if $Q \in STA$ and $Q \approx P$, where STA denotes the set of all standard (i.e,. self-needed, or equivalently, self-essential) reduction, and call Q a *standard variant* of P. In particular, $ST(P) \in STV(P)$.

Theorem 2.9 (Standardization) For any finite reduction P in a stable non-duplicating DRS, $ST(P)$ is a standard reduction Lévy-equivalent to P.

2.2 Families in Affine SDRSs

In this section we recall construction of DFSs from non-duplicating SDRSs from [KG96a].

Below, $FAM(P)$ (resp. $SFAM(P)$) denotes the set of *zig-zag classes* whose member (resp. P-needed) redexes are contracted in P, in an ASDRS.

Proposition 2.10 Let P and Q be co-initial finite reductions in an affine zig-zag DFS. Then $P \approx Q$ iff $SFAM(P) = SFAM(Q)$. Further, $SFAM(P) = FAM(ST(P)) = \cap_{N \approx P} FAM(N)$. Consequently, if $P, Q \in STA$, then $P \approx Q$ iff $FAM(P) = FAM(Q)$ (since in that case $P = ST(P)$ and $Q = ST(Q)$).

Theorem 2.11 (Affine Extraction) Every redex Pv in an ASDRS has a zig-zag related redex $P'v'$ in *canonical* form, meaning that $P' \in STA$ and for any $P''v'' \simeq_z Pv$, $\exists N' : P'' \approx P' + N' \wedge v'' = v'/N$. For any other canonical form P^*v^* of Pv, $P' \approx P^*$ and $v' = v^*$.

Theorem 2.12 (Affine Zig-zag Families) Let $\mathcal{R}$ be a non-duplicating SDRS. Then $\mathcal{F}_{\mathcal{R}} = (\mathcal{R}, \simeq_z, \hookrightarrow_z)$ is a zig-zag DFS, called the *corresponding* ZDFS of $\mathcal{R}$.

3 Deterministic Permutation and Erasure Event Structures

In this section, we introduce Deterministic Permutation and Erasure Event Structures, DPESs and DEESs, which extend conflict-free Prime Event Structures by an axiomatized permutation-equivalence and erasure relations, respectively, and establish an equivalence between them.

A *Prime Event Structure* (PES) [Win80] is a triple $\mathcal{E} = (E, Con, \leq)$, where E is a set of *events*, ranged over by $e, e_1, \ldots$; the *consistency predicate* Con is a non-empty set of finite 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 only consider *conflict-free* or *deterministic* PESs, where no event can prevent others from occurring, and therefore the consistency predicate is the set of all finite subsets of E, $Fin(E)$, and will be omitted. Finite *configurations* of $\mathcal{E}$ are finite *left-closed subsets* $\alpha, \beta, \ldots$ of E, i.e., subsets $\mathcal{L}(E) = \{\alpha \subseteq Fin(E) \mid e \in \alpha \wedge e' < e \Rightarrow e' \in \alpha\}$. We only consider finite configurations, and will omit 'finite'.

Below, for better readability, we write $\alpha + e$ for $\alpha \cup \{e\}$, and write $\alpha - e$ for $\alpha \setminus \{e\}$. Further, we define $\lceil e^{\leq} \rceil = \{e' \mid e' \leq e\}$, $\lceil e^{<} \rceil = \{e' \mid e' < e\}$, $\lceil e^{\geq} \rceil = \{e' \mid e \geq e'\}$ and $\lceil e^{>} \rceil = \{e' \mid e' > e\}$.

Definition 3.1 A *Deterministic Permutation Event Structure* (DPES) is a triple $\mathcal{P} = (E, \leq, \approx)$, where $(E, \leq)$ is a conflict-free PES and $\approx \subseteq \mathcal{L}(E) \times \mathcal{L}(E)$ is an equivalence relation, called *Lévy* or *permutation-equivalence*, satisfying the following axioms, where $\alpha, \beta, \gamma \in \mathcal{L}(E)$:

- [P0] $\forall e \in E : \lceil e^{<} \rceil \not\approx \lceil e^{\leq} \rceil$;
- [P1] $\alpha \approx \beta \wedge \alpha \cap \beta \subseteq \gamma \subseteq \alpha \cup \beta \Rightarrow \alpha \approx \gamma$;
- [P2] $\alpha \approx \beta \wedge \alpha + e, \beta + e \in \mathcal{L}(E) \Rightarrow \alpha + e \approx \beta + e$;
- [P3] $\alpha \approx \alpha - e \wedge e < e' \wedge \alpha + e' \in \mathcal{L}(E) \Rightarrow \alpha \approx \alpha + e'$;

DEESs extend *Deterministic Computation Structures* (DCS) [GK96] by further erasure axioms to enable a fully adequate treatment of erasure.

Definition 3.2 A *Deterministic Erasure Event Structure* (DEES) is a triple $\mathcal{C} = (E, \leq, \rhd)$, where $(E, \leq)$ is a conflict-free PES and $\rhd \subseteq \mathcal{L}(E) \times E$ is *inessentiality* or *erasure* relation, satisfying the following axioms, where $\alpha, \beta \in \mathcal{L}(E)$:

- [E0] $\forall e \in E : \emptyset \not\rhd e$;
- [E1] $\alpha \rhd e \wedge \alpha \subseteq \beta \Rightarrow \beta \rhd e$;
- [E2] $\alpha \rhd e' \wedge \alpha \rhd e \wedge \alpha - e' \in \mathcal{L}(E) \Rightarrow \alpha - e' \rhd e$;
- [E3] $\alpha \rhd e \wedge e < e' \Rightarrow \alpha \rhd e'$;
- [E4] $\alpha \cup \lceil e^{<} \rceil \rhd e \Rightarrow \alpha \rhd e$.

We read $\alpha \rhd e$ as: 'e is α-*inessential*'.

We may write $\alpha \rhd X$ if $\forall e \in X : \alpha \rhd e$. Sometimes we write $\mathcal{L}(\mathcal{P})$ for $\mathcal{L}(E)$ if E is the domain of a DPES $\mathcal{P}$, and similarly for $\mathcal{L}(\mathcal{C})$. Next we define translations between DPESs and DEESs.

Definition 3.3 For a DPES $\mathcal{P} = (E, \leq, \approx)$, we define an *erasure* relation $\rhd_{\approx} \subseteq \mathcal{L}(E) \times E$ and the corresponding structure $\mathcal{C}_{\mathcal{P}} = (E, \leq, \rhd_{\approx})$ as follows:

- $\alpha \rhd_{\approx} e$ iff $\alpha \cup \lceil e^{\leq} \rceil \approx (\alpha \cup \lceil e^{\leq} \rceil) \setminus \lceil e^{\geq} \rceil$.

In particular, when $e \in \alpha$ and $\alpha - e \in \mathcal{L}(E)$,

- $\alpha \rhd_{\approx} e$ iff $\alpha \approx \alpha - e$.

Definition 3.4 For a DEES $\mathcal{C} = (E, \leq, \rhd)$, we define an equivalence relation $\approx_{\rhd} \subseteq \mathcal{L}(E) \times \mathcal{L}(E)$ and the corresponding structure $\mathcal{P}_{\mathcal{C}} = (E, \leq, \approx_{\rhd})$ as follows:

- $\alpha \approx_{\rhd} \beta$ iff $ST_{\rhd}(\alpha) = ST_{\rhd}(\beta)$, where $ST_{\rhd}(\alpha) = \{e \in \alpha \mid \alpha \not\rhd e\}$ is the set of α-*essential* events of α.

(The subscript $\rhd$ in $ST_{\rhd}(\alpha)$ will often be omitted.)

$ST(\alpha)$ is called the *standard variant* of α. It is easy to show that $\alpha \in \mathcal{L}(E) \Rightarrow ST(\alpha) \in \mathcal{L}(E)$.

A close inspection of the above definitions shows that the DPES axiom [P0] is a counterpart of the

DEES axioms [E0] and [E4], and similarly for [P3] and [E3]. Further, the combination of [P1] and [P2] have the same effect as that of [E1] and [E2]. [E4] does not have a 'direct' counterpart among DPES axioms, but it corresponds to the fact that, according to Definition 2.7, a redex with history Pv is Q-inessential iff v is Q/P-inessential. Similarly, [P0] corresponds to Lemma 2.8.(2).

We need the following characterization of $\approx_\rhd$ via $\rhd$ and $\approx$ via $\rhd_\approx$, which is not difficult to prove from the erasure and permutation axioms.

Proposition 3.5 $\forall \alpha, \beta, \gamma \in \mathcal{L}(\mathcal{C}) : \alpha \cap \beta \subseteq \gamma \subseteq \alpha \cup \beta, \alpha \approx_\rhd \beta \Leftrightarrow ST(\alpha) = ST(\beta) \Leftrightarrow \gamma \rhd \alpha \cup \beta \setminus \alpha \cap \beta$.

Proposition 3.6 In a DPES $\mathcal{P} = (E, \leq, \approx)$, $\alpha \approx \beta$ iff $ST_{\rhd_\approx}(\alpha) = ST_{\rhd_\approx}(\beta)$.

Theorem 3.7 For any DEES $\mathcal{C} = (E, \leq, \rhd)$, $\mathcal{P}_\mathcal{C} = (E, \leq, \approx_\rhd)$ is a DPES.

Proof We need to show that $\approx_\rhd$ satisfies the permutation axioms [P0] – [P3]. We only prove [P2] here.

[P2]: Let $\alpha \approx_\rhd \beta$ and $\alpha + e, \beta + e \in \mathcal{L}(E)$. By Definition 3.4, $ST(\alpha) = ST(\beta)$. Assume first that $ST(\alpha) \rhd e$. Then $\alpha + e \rhd e$ by [E1] and $ST(\alpha + e) = ST(\alpha)$ by Proposition 3.5. Similarly, $ST(\beta + e) = ST(\beta)$. Hence, $ST(\alpha + e) = ST(\beta + e)$, implying by Definition 3.4 that $\alpha + e \approx_\rhd \beta + e$. Now let $ST(\alpha) \not\rhd e$. Then, by [E3], $\forall e' \in \alpha \setminus ST(\alpha) : e' \not< e$. Thus $\alpha + e \in \mathcal{L}(E) \Rightarrow ST(\alpha) + e \in \mathcal{L}(E)$. Since $\alpha + e \rhd \alpha + e \setminus (ST(\alpha) + e)$, we have by Proposition 3.5 that $ST(\alpha + e) = ST(ST(\alpha) + e)$, thus $\alpha + e \approx_\rhd ST(\alpha) + e$ by Definition 3.4. Similarly, $\beta + e \approx_\rhd ST(\beta) + e$, implying $\alpha + e \approx_\rhd \beta + e$ (since $ST(\alpha) = ST(\beta)$).

Theorem 3.8 For any DPES $\mathcal{P} = (E, \leq, \approx)$, $\mathcal{C}_\mathcal{P} = (E, \leq, \rhd_\approx)$ is a DEES.

Proof We need to show that $\rhd_\approx$ satisfies the erasure axioms [E0] – [E4]. [E0] follows from Definition 3.3 and [P0], and [E4] is immediate from Definition 3.3. Proofs of the other axioms are not too long, but are tedious to include here.

Theorem 3.9 (Equivalence of DPESs and DEESs)

(1) For any DEES $\mathcal{C} = (E, \leq, \rhd)$, $\mathcal{C}_{\mathcal{P}_\mathcal{C}} = \mathcal{C}$.

(2) For any DPES $\mathcal{P} = (E, \leq, \approx)$, $\mathcal{P}_{\mathcal{C}_\mathcal{P}} = \mathcal{P}$.

Proof Note that the translations of DEESs into DPES and vice versa does not effect the sets of events and the causality relation. Let $\mathcal{C}_{\mathcal{P}_\mathcal{C}} = (E, \leq, \rhd')$ and $\mathcal{P}_{\mathcal{C}_\mathcal{P}} = (E, \leq, \approx')$, and let us show that $\rhd = \rhd'$ and $\approx = \approx'$.

(1) By [E4], [E1] and [E3], $\alpha \rhd e$ iff $\alpha \cup \lceil e^{\leq} \rceil \rhd e$ iff $\alpha \cup \lceil e^{\leq} \rceil \rhd \lceil e^{\geq} \rceil$. By Proposition 3.5, the latter holds iff $ST(\alpha \cup \lceil e^{\leq} \rceil) = ST(\alpha \cup \lceil e^{\leq} \rceil \setminus \lceil e^{\geq} \rceil)$, which, by Definition 3.4 means that $\alpha \cup \lceil e^{\leq} \rceil \approx_\rhd \alpha \cup \lceil e^{\leq} \rceil \setminus \lceil e^{\geq} \rceil$, meaning $\alpha \rhd' e$ by Definition 3.3.

(2) By Proposition 3.6 and Definition 3.4, $\alpha \approx \beta$ iff $ST_{\rhd_\approx}(\alpha) = ST_{\rhd_\approx}(\beta)$ iff $\alpha \approx' \beta$.

4 Relating Transition and Event models

We now define translations of ASDRSs and AZDFSs into DPESs and DEESs, define the converse translations, and show that the translations commute, implying equivalence of the four computational models.

4.1 Translation of affine SDRSs and ZDFSs into DPESs and DEESs

Definition 4.1 To an AZDFS $\mathcal{F} = (\mathcal{R}, \simeq, \hookrightarrow)$, we associate the DPES $\mathcal{P}_\mathcal{F} = (E_\mathcal{F}, \leq, \approx)$, and the DEES $\mathcal{C}_\mathcal{F} = (E_\mathcal{F}, \leq, \rhd)$, where

- $E_\mathcal{F} = FAM(\mathcal{F})$, the set of all families of $\mathcal{F}$;
- $\phi < \psi$ iff $\phi \hookrightarrow \psi$;
- $\alpha \approx \beta$ iff $\forall \phi \in \alpha \cup \beta \setminus \alpha \cap \beta$, $\exists P : \emptyset \twoheadrightarrow s : FAM(P) \subseteq \alpha \cap \beta$ and P discards a canonical element of ϕ;
- $\alpha \rhd \phi$ iff $\exists P : \emptyset \twoheadrightarrow s : FAM(P) \subseteq \alpha$ and P discards a canonical element of ϕ.

In order to prove the correctness of Definition 4.1, we need, among others, the following lemma, which is obtained using results in subsections 2.1 and 2.2.

Lemma 4.2 Let $Q : t \twoheadrightarrow s$, $P : t \twoheadrightarrow o$, $v \in o$, let Pv be Q-inessential, and let $Q' \in STV(Q)$, in an ASDRS. Then Q' discards Pv.

Lemma 4.3 Let $P \in STA$, $FAM(P) = F_1 \cup F_2$, $F_1 \cap F_2 = \emptyset$ and $\forall \phi \in F_2, \forall \psi \in F_1 : \phi \not\hookrightarrow \psi$. Then $\exists Q \in STV(P) : Q = Q_1 + Q_2, FAM(Q_i) = F_i$.

Proof Let u be the latest step in P such that $\phi = Fam(u) \in F_2$ and it has a next step v with $\psi = Fam(v) \in F_1$. Then $\phi \not\hookrightarrow \psi$ by the hypothesis, and u does not create v by [creation]. Hence u and v can be permuted, yielding again a standard reduction by Lemma 2.8.(4). The resulting reduction can be transformed similarly until all steps in F_2 are pushed to the end. The transformation terminates as it does not change the length of reductions.

Lemma 4.4 For any AZDFS $\mathcal{F}$, $\mathcal{C}_\mathcal{F}$ is a DEES.

Proof We need to show that $\mathcal{C}_\mathcal{F} = (E_\mathcal{F}, \hookrightarrow, \rhd)$ satisfies the erasure axioms. We only prove [E4].

[E4]: Let $\alpha \cup \lceil \phi^{<} \rceil \rhd \phi$. By Definition 4.1, there is a Q such that $FAM(Q) \subseteq \alpha \cup \lceil \phi^{<} \rceil$ and Q discards

a canonical element Pv of ϕ. By Proposition 2.10 and Lemma 4.2, Q can be chosen standard. Let $F_2 = FAM(Q) \cap (\lceil \phi^< \rceil \setminus \alpha)$ and $F_1 = FAM(Q) \setminus F_2$. Since $\alpha \in \mathcal{L}(E)$, $\forall \phi \in F_2, \psi \in F_1 : \phi \not\hookrightarrow \psi$. Hence, by Lemma 4.3, $\exists Q \in STV(P) : Q' = Q_1 + Q_2, FAM(Q_i) = F_i$. By Lemma 4.2, Q' discards Pv, and $Q'/P = Q_1/P$, since $\lceil \phi^< \rceil \subseteq FAM(P)$. Hence Q_1 discards Pv and $FAM(Q_1) \subseteq \alpha$, implying $\alpha \rhd \phi$.

It is an immediate consequence of Definitions 4.1 and 3.4, Proposition 3.6 and the above lemma that for any AZDFS $\mathcal{F}$, $\mathcal{P}_\mathcal{F} = \mathcal{P}_{\mathcal{C}_\mathcal{F}}$. By combining this with Theorems 3.7 and 3.9, we get the following theorem.

Theorem 4.5 For any AZDFS $\mathcal{F}$, $\mathcal{C}_\mathcal{F}$ is a DEES, $\mathcal{P}_\mathcal{F}$ is a DPES, $\mathcal{P}_\mathcal{F} = \mathcal{P}_{\mathcal{C}_\mathcal{F}}$ and $\mathcal{C}_\mathcal{F} = \mathcal{C}_{\mathcal{P}_\mathcal{F}}$.

Since ASDRSs are AZDFSs by Theorem 2.12, from the above results we immediately get interpretations of ASDRSs into DPESs and DEESs.

4.2 Translation of DPESs and DEESs into stable DRSs and DFSs

The following translation of DPESs and DEESs into stable DRSs resembles to the representation of events in a PES $(E, \leq)$ as prime intervals of the configuration domain $\mathcal{L}(E)$, used for construction of PESs from stable families of configurations in [Win86/89].

Definition 4.6 To a DPES $\mathcal{P} = (E, \leq, \approx)$ (resp. a DEES $\mathcal{C} = (E, \leq, \rhd)$) we associate an SDRS $\mathcal{R}_\mathcal{P}$ (resp. $\mathcal{R}_\mathcal{C}$) as follows:

- The terms of $\mathcal{R}_\mathcal{P}$ ($\mathcal{R}_\mathcal{C}$) are Lévy-equivalence classes $\langle\alpha\rangle_L, \langle\beta\rangle_L, \ldots$ of configurations of $\mathcal{P}$ ($\mathcal{C}$);
- The reduction relation of $\mathcal{R}_\mathcal{P}$ ($\mathcal{R}_\mathcal{C}$) consists of sets of pairs $u = (\langle\alpha\rangle_L, e)$, where $\alpha, \alpha + e \in \mathcal{L}(E)$ and $\alpha \not\approx \alpha + e$ (resp. $\alpha \not\rhd e$). Pairs $u = (\langle\alpha\rangle_L, e)$ such that $\alpha \approx \alpha + e$ ($\alpha \rhd e$) are identified with the empty redex $\emptyset = (\langle\alpha\rangle_L, \emptyset)$ in $\langle\alpha\rangle_L$.
- The residual relation is defined as follows: if $u = (\langle\alpha\rangle_L, e)$ and $v = (\langle\alpha\rangle_L, e')$, then $u/v = (\langle\alpha + e'\rangle_L, e)$.

It is easily seen that $\mathcal{R}_\mathcal{P} = \mathcal{R}_{\mathcal{C}_\mathcal{P}}$ and $\mathcal{R}_\mathcal{C} = \mathcal{R}_{\mathcal{P}_\mathcal{C}}$. In [GK96], a similar translation of DCSs into SDRSs is given, and its correctness implies also the correctness of Definition 4.6, i.e., $\mathcal{R}_\mathcal{P}$ and $\mathcal{R}_\mathcal{C}$ are indeed SDRSs.

Note that initial reductions in $\mathcal{R}_\mathcal{P}$ and $\mathcal{R}_\mathcal{C}$ can be represented as event sequences $e_1, \ldots, e_n$ respecting the causal order – e_i represents the (contraction of) *non-empty* redex $(\{e_1, \ldots, e_{i-1}\}, e_i)$. That is, reductions are (some of) left-closed enumerations of configurations $\alpha \in \mathcal{L}(E)$, and will be denoted by $[\alpha]$. As we will see below, not all left-closed enumerations of a configuration represent a reduction. Such enumerations will be called *quasi-reductions*. Quasi-reductions are defined similarly in any DRS – they are 'reductions containing empty steps'.

Definition 4.7 To a DPES $\mathcal{P} = (E, \leq, \approx)$, we associate a DFS $\mathcal{F}_\mathcal{P} = (\mathcal{R}_\mathcal{P}, \simeq_\mathcal{P}, \hookrightarrow_\mathcal{P})$ as follows:

- $\mathcal{R}_\mathcal{P}$ is the DRSs defined in Definition 4.6.
- $\forall e \in E$, we define the $\simeq_\mathcal{P}$-family ϕ_e corresponding to e by $\phi_e = \{([\alpha], e) \mid \alpha + e \in \mathcal{L}(\mathcal{P}) \wedge (\langle\alpha\rangle_L, e) \neq \emptyset\}$. That is, $([\alpha], e) \simeq_\mathcal{P} ([\beta], e')$ iff $e = e'$, $\alpha + e, \beta + e \in \mathcal{L}(\mathcal{P}), (\langle\alpha\rangle_L, e) \neq \emptyset, (\langle\beta\rangle_L, e) \neq \emptyset$.
- Define $\hookrightarrow_\mathcal{P}$ by: $\phi_e \hookrightarrow_\mathcal{P} \phi_{e'}$ iff $e < e'$.

One easily shows that translations $\mathcal{P} \to \mathcal{F}_\mathcal{P}$ and $\mathcal{P} \to \mathcal{R}_\mathcal{P}$ commute with the zig-zag embedding $\mathcal{R} \to \mathcal{F}_\mathcal{R}$, i.e., $\mathcal{F}_{\mathcal{R}_\mathcal{P}} = (\mathcal{R}_\mathcal{P}, \simeq_z, \hookrightarrow_z) = (\mathcal{R}_\mathcal{P}, \simeq_\mathcal{P}, \hookrightarrow_\mathcal{P}) = \mathcal{F}_\mathcal{P}$, implying in particular the correctness of Definition 4.7.

Theorem 4.8 For any DPES $\mathcal{P}$, $\mathcal{F}_\mathcal{P}$ is an AZDFS.

4.3 Equivalence of the four models

For any of the four models, there is an obvious concept of *isomorphism*, $\rightleftharpoons$. For example, DFSs $\mathcal{F} = (\mathcal{R}, \simeq, \hookrightarrow)$ and $\mathcal{F}' = (\mathcal{R}', \simeq', \hookrightarrow')$ are isomorphic if there is an isomorphism between the underlying reduction graphs of $\mathcal{R}$ and $\mathcal{R}'$ respecting the residual, family, and contribution relations in $\mathcal{F}$ and $\mathcal{F}'$.

Theorem 4.9
(1) For any DPES $\mathcal{P}$, $\mathcal{P} \rightleftharpoons \mathcal{P}_{\mathcal{F}_\mathcal{P}}$.
(2) For any AZDFS $\mathcal{F}$, $\mathcal{F} \rightleftharpoons \mathcal{F}_{\mathcal{P}_\mathcal{F}}$.
Proof Routine, from Definition 4.1 and Definition 4.7.

This theorem, combined with the results of previous sections, implies equivalence of the considered four transition and event models.

5 Configuration Domains

Let us call a left-closed enumeration (i.e, respecting the causality relation) $e_1, \ldots, e_k$ of a configuration α a *reduction* or an *α-reduction* if no event e_i in the sequence is *vacuous*, i.e., $\forall i : \{e_1, \ldots, e_{i-1}\} \not\rhd e_i$. Any other left-closed enumeration of α will be called a *quasi-reduction*. An α-reduction will be denoted by $[\alpha]$. Thus reductions in a DPES $\mathcal{P}$ precisely correspond to initial reductions in the corresponding ASDRS $\mathcal{R}_\mathcal{P}$, and the use of notation $[\alpha]$ both for reductions in $\mathcal{P}$ and $\mathcal{R}_\mathcal{P}$ should not be confusing.

Note that, unlike Prime Event Structures, not every configuration α in a DEES has a reduction. For

example, consider the DEES consisting of four events $\{e_1, e_2, e_1', e_2'\}$, ordered as follows: $e_1 < e_1'$ and $e_2 < e_2'$, and let $\rhd$ be given by: $\alpha \rhd e_1'$ iff $e_2 \in \alpha$, and $\alpha \rhd e_2'$ iff $e_1 \in \alpha$. Then $\{e_1, e_2, e_1', e_2'\}$ is a configuration, but any of its is a left-closed enumerations, such as e_1, e_2, e_1', e_2', ends either with e_1' or with e_2', and we have $\{e_1, e_2, e_2'\} \rhd e_1'$ and $\{e_1, e_2, e_2'\} \rhd e_1'$. This is not surprising as the vacuous events in left-closed sequences correspond to empty reductions in the corresponding ASDRS, performed in the 'garbage' (erased redexes). Thus quasi-reductions in a DPES/DEES correspond to quasi-reductions in the corresponding ASDRS. However, we have the following lemma.

Lemma 5.1 Let $\alpha \in \mathcal{L}(\mathcal{C})$, in a DEES $\mathcal{C}$. Then any left-closed enumeration of $ST(\alpha)$ is a reduction.
Proof Let $e_0, \ldots, e_n$ be a left-closed enumeration of $ST(\alpha)$. If there was an i such that $e_0, \ldots, e_{i-1} \rhd e_i$, then $\alpha \rhd e_i$ by [E1], contradicting $e_i \in ST(\alpha)$.

Definition 5.2 Let $\mathcal{P}$ be a DPES, $\mathcal{L}^{\approx}(\mathcal{P}) = \mathcal{L}(\mathcal{P})/_{\approx} = \{\langle\alpha\rangle_L \,|\, \alpha \in \mathcal{L}(\mathcal{P})\}$, and let $\mathcal{L}^{\trianglelefteq}(\mathcal{P}) = (\mathcal{L}^{\approx}(\mathcal{P}), \trianglelefteq)$ be a partial order defined by $\langle\alpha\rangle_L \trianglelefteq \langle\beta\rangle_L$ iff $\exists\alpha' \in \langle\alpha\rangle_L, \exists\beta' \in \langle\beta\rangle_L : \alpha' \subseteq \beta'$.

From the above comparison of (quasi)-reductions in DPESs and ASDRSs, we have the following fact:

Proposition 5.3 $\langle P\rangle_L \trianglelefteq \langle Q\rangle_L$ in an ASDRS $\mathcal{R}$ iff $\langle FAM(P)\rangle_L \trianglelefteq \langle FAM(Q)\rangle_L$ in $\mathcal{P}_{\mathcal{R}}$.

Definition 5.4 Let $\mathcal{R}$ be an ASDRS, $\mathcal{L}^{\approx}(\mathcal{R}) = \mathcal{L}(\mathcal{R})/_{\approx} = \{\langle P\rangle_L \,|\, P \in \mathcal{L}(\mathcal{R})\}$, where $\mathcal{L}(\mathcal{R})$ is the set of initial reductions in $\mathcal{R}$, and let $\mathcal{L}^{\trianglelefteq}(\mathcal{R}) = (\mathcal{L}^{\approx}(\mathcal{R}), \trianglelefteq)$, where $\trianglelefteq$ is Lévy's embedding relation.

Theorem 5.5 Let $\mathcal{R}$ be an ASDRS and let $\mathcal{P} = \mathcal{P}_{\mathcal{R}}$ be its corresponding DPES. Then $f : \langle P\rangle_L \to \langle FAM(P)\rangle_L$ is the isomorphism between $\mathcal{L}^{\trianglelefteq}(\mathcal{R})$ and $\mathcal{L}^{\trianglelefteq}(\mathcal{P})$, with the inverse $f^{-1} : \langle\alpha\rangle_L \to \langle[ST(\alpha)]\rangle_L$. Moreover, (quasi-)reductions in $\langle P\rangle_L$ are in one-to-one correspondence with these in $\langle FAM(P)\rangle_L$.
Proof By Proposition 5.3 and Lemma 5.1.

Note that $\mathcal{L}^{\trianglelefteq}(\mathcal{P})$ is not a lattice since $\mathcal{L}^{\trianglelefteq}(\mathcal{R})$ is not. However, depending on the set of 'results' we are interested in, we can construct a (distributive) lattice on configurations in $\mathcal{P}$ reflecting growth of information relative to that goal. For example, if we are interested in fair computations, we can order configurations by defining $\alpha \preceq \beta$ iff $NE(\alpha) \subseteq NE(\beta)$, where $NE(\alpha)$ is the set of *fairly-needed* events of α. Here an event is fairly-needed in $\mathcal{P}$ if it belongs to every, possibly infinite, configuration in the maximal element of the ideal completion of $\mathcal{L}^{\trianglelefteq}(\mathcal{P})$. Fairly-needed redexes in $\mathcal{R}$ coincide with *Maranget-needed* or *essential* redexes, and are contracted in all fair reductions in $\mathcal{R}$, which are maximal w.r.t. $\trianglelefteq$ [GK94]. Note that $\preceq$ induces an equivalence relation which is different from the permutation-equivalence. Construction of such 'relativized' information domains from configurations in DPESs will be addresses in detail in a separate paper.

6 Event Structure semantics for stable DRSs

Now we come to constructing event semantics for orthogonal rewrite systems. Our interpretations is based on interpreting redex-families as events, so we assume that if the system is duplicating, then it comes with a notion of redex-family (defined via labelling, extraction or zig-zag [Lév78, Lév80, Mar91, AL93, Oos96]) satisfying our family axioms.

Recall that a multi-step reduction in a DFS is a *(complete) family-reduction* if each multi-step is a complete development of the maximal set of redexes of some family. it is immediate that for any DFS $\mathcal{F} = (\mathcal{R}, \simeq, \hookrightarrow)$, $\mathcal{E}_{\mathcal{F}} = (FAM(\mathcal{F}), \underline{\hookrightarrow})$ is a PES [GK96]. Note however that some of the events in $FAM(\mathcal{F})$ may never occur in reduction sequences of $\mathcal{E}_{\mathcal{F}}$ that correspond to family-reductions in $\mathcal{F}$. [1] As argued in the introduction, we want events in the corresponding DPES of $\mathcal{F}$ to match precisely the families contracted in family-reductions in $\mathcal{F}$. Therefore, we first associate an ASDRS $\mathcal{F}_l$ to a DFS $\mathcal{F}$, called the *linearization* of $\mathcal{F}$, and interpret $\mathcal{F}$ in the DEES or DPES corresponding to $\mathcal{F}_l$. The reduction steps in $\mathcal{F}_l$ correspond to family-reduction multi-steps of $\mathcal{F}$.

Definition 6.1 Let $\mathcal{F} = (\mathcal{R}, \simeq, \hookrightarrow)$ be a DFS. The *linearization* of $\mathcal{F}$ is the ASDRS $\mathcal{F}_l = (A, /)$, where

- the branches of the reduction graph of the underlying ARS A are family-reductions starting from $\emptyset$, each edge (i.e., a reduction step) being a multi-step contracting a family of redexes.
- the residual relation $/$ is defined as follows: let U and V be complete sets of redexes in two families, in a term s, and let $U : s \to o$ be the multi-step contacting U. Then V/U is the multi-step $o \to e$ contracting all members of the set V/U in a multi-step.

[1] For example, consider a CRS ([Klo80]) R with rules $\{F[x]Z(x) \to Z(Z(x)), H(x) \to a, G(a) \to b\}$, restricted to the reduction graph of the term $t = F[x]H(G(x))$. Let $P : F[x]H(G(x)) \to H(G(H(G(x)))) \to H(G(a))$ and $v = G(a)$. Then the redex Pv has a corresponding event in the corresponding PES $(FAM(R), \underline{\hookrightarrow})$, but no family-reduction can create redexes in $Fam(Pv)$.

It is easy to check that $\mathcal{F}_l$ in the above definition is indeed an ASDRS. One first checks that if U and V are complete sets of redexes in two different families in a term s and $s \xrightarrow{U} o$, then V/U forms a complete family too, in o. Obviously, $V/V = \emptyset$, and every family in o has at most one ancestor family in s. Finally, [acyclicity] and [stability] for $\mathcal{F}_l$ follow immediately from the Acyclicity Lemma and Stability Lemma, respectively. From the correctness of Definition 6.1 and Theorem 2.12, we have the following theorem.

Theorem 6.2 For any DFS $\mathcal{F}$, $\mathcal{F}_l$ is an AZDFS.

The extraction procedure [KG96a] in the linearization R_l of an orthogonal (possibly duplicating) rewrite system R, first or higher order, which comes with the labelling concept of family [Lév78, Mar91, AL93, Oos96], corresponds to 'family-extraction' in family-reductions in R, eliminating all family-steps not contributing to a family. (So family-extraction is different from extraction in that the latter eliminates particular redexes in a history, which need not be a family-reduction.) Two families in different family-reductions can be family-extracted to the same canonical representative iff the families have the same label (since the label of a family is built exactly from the labels of contributor families, and extraction preserves the labels). Here a canonical representative of a family is one whose history is a family-reduction contracting only contributor families; its existence and uniqueness up to the order of contraction of causally independent families follows from the Affine Extraction theorem. Since labelling and family-extraction families coincide in orthogonal rewrite systems and zig-zag and extraction families coincide in ASDRSs [KG96a], families contracted in family-reductions in R are in one-to-one correspondence with zig-zag families in R_l.

By combining the translation (linearization) of DFSs into AZDFS with the translations of the latter into DPESs and DEESs, we achieve our objective of defining an adequate event style semantics for orthogonal rewrite systems.

7 Conclusions

We have established an equivalence between deterministic stable operational (ASDRSs) and domain-theoretic (DPESs and DEESs) models of computation, and based on this, constructed a fully adequate event style concurrent semantics for orthogonal rewrite systems. The correspondence between Event Structures and the denotational semantics is well studied [Win86/89], and we hope that our results contribute to better understanding of relationship between operational and denotational semantics of sequential and concurrent languages.

For example, the theory of normalization by neededness relative to *stable sets of terms* in ASDRSs [GK96] has a counterpart in DPESs, thus implying a concept of *stable sets of configurations* in DPESs, different from the concept of *stable families of configurations* in [Win86/89].[2] The concept of *stability* of an Event Structure $(E, \vdash)$ (with an *enabling* relation $\vdash$ instead of a causal order $\leq$) [Win86/89] becomes a special case of our concept of stable sets, as it is trivial to check that $(E, \vdash)$ is stable iff for any $e \in E$, so is the set of all configurations containing e. Study of precise relationship between syntactic and semantic concepts of stability and its relationship with sequentiality (neededness) goes beyond the scope of this paper.

We think that our axiomatization of permutation equivalence is interesting on its own right from the lattice theory point of view. Definitions of permutation equivalence without using a residual calculus have been studied before [LM94], but they still use the syntax of the rewrite system essentially (axioms do depend on the syntax).

We expect that the theory developed here for conflict-free systems can be extended to the general case, so that higher order process calculi in which processes can be passed and erased can be modeled too.

References

[AL93] Asperti A., Laneve C. Interaction Systems I: The theory of optimal reductions. MSCS 11:1-48, 1993.

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

[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.

[2] A set $\mathcal{S}$ of configurations in a DPES $\mathcal{P}$ is *stable* if it is closed under Lévy-equivalence, $\alpha \in \mathcal{S} \wedge \beta \in \mathcal{S} \Rightarrow \alpha \cap \beta \in \mathcal{S}$, and $(\alpha \in \mathcal{S} \wedge \beta \in \mathcal{L}(\mathcal{P}) \wedge \nexists \gamma \in S : \gamma \subseteq \beta) \Rightarrow \alpha \cup \beta \in \mathcal{S}$. Recall that stability of $\mathcal{S}$ is a sufficient and necessary condition for the $\mathcal{S}$-needed strategy to be $\mathcal{S}$-normalizing [GK94, GK96].

[CK95] Clark D., Kennaway R. Some properties of non-orthogonal term graph rewriting. ENTCS 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.

[GK94] Glauert J.R.W., Khasidashvili Z. Minimal and optimal relative normalization in Orthogonal Expression Reduction Systems. Report SYS-C94-06, UEA, Norwich, 1994.

[GK96] Glauert J.R.W., Khasidashvili Z. Relative normalization in deterministic residual structures. In: Proc. of CAAP'96, Springer LNCS, vol. 1059, H. Kirchner, ed. 1996, p. 180-195.

[GLM92] Gonthier G., Lévy J.-J., Melliès P.-A. An abstract Standardisation theorem. In: Proc. of LICS'92, Santa Cruz, California, 1992, p. 72-81.

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

[HL91] 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.

[KKSV93] Kennaway J. R., Klop J. W., Sleep M. R, de Vries F.-J. Event structures and orthogonal term graph rewriting. In [SPE93].

[KG96] Khasidashvili Z., Glauert J. R. W. Discrete normalization and Standardization in Deterministic Residual Structures. In: Proc. of ALP'96, Springer LNCS, vol. 1139, M. Hanus, M. Rodríguez-Artalejo, eds. 1996, p.135-149.

[KG96a] Khasidashvili Z., Glauert J.R.W. Zig-zag and extraction families in non-duplicating stable Deterministic Residual Structures. Submitted. (ftp://ftp.sys.uea.ac.uk/pub/zurab/LinZig.ps.gz)

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

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

[Lan94] Laneve C. Distributive evaluations of λ-calculus, Fundamenta Informaticae, 20(4):333 – 352, 1994.

[LM94] Laneve C., Montanari U. Axiomatizing permutation equivalence. MSCS, 6:219 - 249, 1994.

[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. of POPL'91, p. 255-269.

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

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

[Oos94] Van Oostrom V. Confluence for Abstract and Higher-Order Rewriting. Ph.D. Thesis, Free University, 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, Free University, Amsterdam, 1996.

[Sch94] Schied G. On relating rewrite systems and graph grammars to event structures. Springer LNCS, vol. 776, Schneider H.-J. Ehrig H., eds, 1994, p. 326-340.

[SPE93] Sleep M. R., Plasmeijer M. J., van Eekelen M. C. J. D., eds. Term Graph Rewriting: Theory and Practice. John Wiley, 1993, p.141-156.

[Sta89] Stark E. W. Concurrent transition systems. TCS 64(3):221-270, 1989.

[Win80] Winskel G. Events in Computation. Ph.D. Thesis, University of Edinburgh, 1980.

[Win86/89] Winskel G. An introduction to Event Structures. Springer LNCS, vol. 354, 1989, p. 364-397. Extended version appears as Cambr. Univ. Comp. Lab. Report n. 95, 1986.