The Action Graph Model as a Link Between Abstract Relation Algebras and Process-Algebraic Specifications*

Thomas F. Gritzner

Institut für Informatik
Technische Universität München
Postfach 20 24 20, Arcisstr. 21
W–8000 München 2 (Germany)

Abstract. Relational algebra in the sense of Schröder and Tarski has been thoroughly studied by mathematicians from the universal algebraic viewpoint and some research has shown many applications of the relational calculus in the fields of graph theory and computer science. This paper addresses the field of process-algebraic languages, which are used for the specification and verification of communicating systems. The construction of a bridge from process algebra to relational algebra is investigated. A process-algebraic language is propounded by adding a parallelism operator to the structure of relational algebra; its semantics is given by the action graph model. The action graph model follows the interpretation of concurrent processes by finite partial orders and, technically, it is constructed by viewing relation algebras as boolean algebras with operators. It establishes a correspondence between relational algebra with parallelism operator and a process algebra, which, in particular, carries an additional combinator playing the rôle of "undo".

1. Introduction

The specification of a programming system is usually achieved by the formal prescription of its behaviour. This is a crucial problem when dealing with distributed systems, which include the nonsequential concept of parallelism. An algebraic approach to the specification and verification of distributed systems is the use of process-algebraic languages, i.e. languages comprising combinators to build processes from other processes [1, 5, 11, 17, 18]. This paper investigates the usage of relational algebra as a process-algebraic language. Relational algebra has a long mathematical tradition. Its present form is established by Tarski inspired by considerations of Schröder. The structure of relation algebras has been thoroughly studied in the past by mathematicians in a universal algebraic setting [9, 12, 14, 15, 16, 26]. But also applications of the relational calculus to graph theory and to computer science have been recognized [10, 23, 24, 25, 28].

*This paper is partially sponsored by the *Sonderforschungsbereich 342 "Werkzeuge und Methoden für die Nutzung paralleler Rechnerarchitekturen"* (DFG).

The interpretation of a specification language for concurrency by a graph-based semantics is a commonly used concept. It is established by fixing a set of atomic actions, which is associated to the graph by the following two ways: actions are assigned either to the *edges* yielding *process graphs* [1, 17, 18] or to the *nodes* yielding *labelled partial orders* [21] or *action structures* [2]. *Labelled transition systems* (e.g. [5]) or *graph programs* [24, 3] are in analogy to process graphs. The graph-based semantics is not used such that a single behaviour is represented by one graph, but the abstraction from the underlying graph structure by the use of congruence classes instead of elements is put into foreground. Two ways of abstractions are mainly considered: to abstract from the selection of the nodes as for labelled partial orders becoming *pomsets* [21] and to abstract from equivalent behaviours as achieved by the graph congruence of *bisimulation* (e.g. [1]) or that of *flow equivalence* (based on the concept of *covering*) [6, 24]. Another modification caused for the labelled partial order interpretation by the concept of *non-determinism* is, on the one hand, the description of non-deterministic behaviour by a *set* of (abstract) graphs [2, 21], on the other hand, the seperation of branching into non-deterministic branching and concurrent branching by the *conflict relation* for *event structures* [27] or by introduction of "and" nodes leading to *and/or trees* (e.g. [20]).

The syntax and semantics of the process-algebraic language derived from relational algebra are defined by the mathematical method of universal algebra (e.g. [9]): syntax is given by a definition of the form *"$\mathfrak{X} = (X, r_1, ..., r_n)$ is an S iff conditions hold"*, i.e. by the definition of a class S of algebras, while semantics results from the construction of a certain model for an S, i.e. a certain member of S. This method simplifies the usual separation of syntax given by a formal language and semantics given by a meaning function. Moreover, it leads to the uniform view of the verification proposition P **sat** $\mathcal{S}$, where P stands for a process term and $\mathcal{S}$ stands for the prescribed process semantics, as $P \subset \mathcal{S}$, where $\subset$ is a presupposed ordering on the considered model expressing the ordering of observations, which is the view as expected from implementing *refinement* propositions (cf. [17, 18, 11]).

The starting point of our investigation is the comparison of the two signatures of ordinary process algebra (e.g. **PA** [1]) and relational algebra (notations adopted from [25]). It yields the following results: on the one hand, for relational algebra the *parallel merge* operator is missing, on the other hand, process algebra does not carry operators like meet $\wedge$, complementation $\overline{\ }$, or transposition $^{\mathsf{T}}$, provided that join $\vee$ is interpreted as *nondeterministic choice* and multiplication $\cdot$ is interpreted as *sequential composition*. Therefore, the paper addresses two problems. The first one deals with the algebraic axiomatisation of parallel composition, which is an operator missing in relational algebra, while the second one concerns the modelling of "undo" inspired by the additional operation of transposition.

In particular, an objective of our paper is to discuss the following inequality called the *unsharpness axiom*:

$$(\dagger) \quad \exists Q, R, S, T : (Q \| R)(S \| T) \neq (QS) \| (RT),$$

It is clearly satisfied by usual models of process algebras that follow both the *interleaving* principle or the *explicit-concurrency* principle. Besides that, it is also closely connected

with results for *heterogeneous abstract relation algebras*. A heterogeneous abstract relation algebra (cf. [25] A.2) is a structure which describes the calculus of heterogeneous relations, i.e. relations whose domain and codomain are not isomorphic, on an abstract level. The desired result is achieved by expressing parallel composition with the help of the concept of *relational products*, the relation-algebraic formalisations of direct products of sets; details are described in the paper. After rewriting parallel merge as proposed, it may be proved that

$$(\ddagger) \quad \forall\, Q,\, R,\, S,\, T\colon\ (Q\|R)(S\|T) \subset (QS)\|(RT)$$

holds, which is the expected result from interleaving semantics of process algebra, too. However, the invalidity of the converse, i.e. the validity of (†) in relation-algebraic style, is still an open question (cf. [25] last two paragraphs of 7.5). So as to establish the syntax of our process-algebraic language we have accepted (†) and (‡) among others as axioms for $\|$, though (†) causes the loss of equational reasoning, which is not essential to our investigations. Note that it is not intended to give a complete set of axioms for parallel composition, but only axioms interesting from the relation-algebraic viewpoint are taken into account.

The semantics of our process-algebraic language uses the concept of *action graphs* which are action structures [2], i.e. *directed acyclic graphs*, carrying only a *finite* set of events, i.e. nodes, and provided with an additional mapping intended for the modelling of "undo". The desired model of a relation algebra with parallelism operator is obtained by putting (abstract) action graphs as atoms representing single processes behaviours and applying the concept of *atom structures* as known from universal algebraic investigations [9, 12] such that a process behaviour is represented by a set of action graphs. Therefore, in contrast to other applications of relational algebra, the action graph model is *not* constructed from the viewpoint of regarding elements of a relation algebra as a mere abstraction of real binary relations, but rather from the combinatorial viewpoint [12, 14, 16], which, moreover, allows to perform the model construction for relation algebras by a computer program as in [7].

The use of relational algebra for process-algebraic specifications leads to an additional operation of transposition and, correspondingly, yields the additional axiom known as the *Schröder equivalences*. As inspired by viewing relations associated to a graph, one possibility of interpretation is to define transposition as graph *conversion* like in [21], but note that [21] does not make use of conversion for process specifications in practice. Another chance to exploit the additional features is to consider the "undo" function as used within the most modern electronic data processing systems: the last action carried out by the user can be undone by selecting the "undo" function, where, after that, the system executes the counterpart of that action to abandon its effect. More precisely, from the system's viewpoint there exists a mapping that maps each action to its counterpart and transposition maps each behaviour to the contrary happening. Furthermore, the *Schröder equivalences*, whose intuitive meaning is not even clear to relation algebraists – only a connection to commutative triangles may be supposed –, maintain a behavioural modelling of "undo" appropriate to some extent.

The paper is organised as follows. The following section contains the definitions of relation algebras with parallelism operator and the corresponding atom structures. Section 3 employs the action graph model, where appropriate operations on action graphs are introduced. The adequacy of the action graph model is shown by two theorems: the first only deals with the model correctness, while the second gives insight into the modelling of "undo". The objective of the fourth section is the investigation of the transition from relation algebras with parallelism operator to heterogeneous abstract relation algebras.

2. Relation Algebras with Parallelism Operator

The structure of relation algebras with parallelism operator is introduced as the syntactic framework of our process-algebraic language. The notation of relation-algebraic operations is adopted from [25], further notations are due to universal algebra (e.g. [9]), and we assume the reader to be familiar with the notion of boolean algebra with operators [12].

2.1 Definition. A structure $\mathfrak{A} = (A, \vee, \wedge, \overline{}, \cdot, {}^{\mathsf{T}}, \|)$ is called a **relation algebra with parallelism operator** iff the following conditions are satisfied:

(i) $(A, \vee, \wedge, \overline{}, \cdot, {}^{\mathsf{T}})$ is a *homogeneous abstract relation algebra*, i.e. the following conditions hold (cf. 4.1):

- $(A, \vee, \wedge, \overline{})$ is a *complete atomic boolean algebra* with universal elements O and L,
- $(A, \cdot)$ is a *monoid* with identity I,
- (S) $\forall Q, R, S \in A\colon\ QR \wedge S = \mathsf{O} \iff Q \wedge SR^{\mathsf{T}} = \mathsf{O} \iff R \wedge Q^{\mathsf{T}}S = \mathsf{O}$;

(ii) $(A, \|)$ is a *semigroup*, and ${}^{\mathsf{T}}$ is an automorphism of $(A, \|)$;

(iii) Concerning $\|$ the following conditions additionally hold:

- (O) $\forall R \in A\colon\ R\|\mathsf{O} = \mathsf{O}\|R = \mathsf{O}$,
- ($*$) $\forall Q, R, S, T \in A\colon\ (Q \vee R)\|(S \vee T) = (Q\|S) \vee (Q\|T) \vee (R\|S) \vee (R\|T)$,
- (†) $\exists\, Q, R, S, T \in A\colon\ (Q\|R) \cdot (S\|T) \neq (Q \cdot S)\|(R \cdot T)$,
- (‡) $\forall\, Q, R, S, T \in A\colon\ (Q\|R) \cdot (S\|T) \subset (Q \cdot S)\|(R \cdot T)$. ◇

2.2 Remark. A relation algebra with parallelism operator is a boolean algebra with three additional operators $\cdot$, ${}^{\mathsf{T}}$, and $\|$, where join $\vee$ is thought to stand for non-deterministic choice, multiplication $\cdot$ for sequential composition, transposition ${}^{\mathsf{T}}$ for "undo" operation, and $\|$ for parallel composition. The axioms of parallel composition $\|$ include essential algebraic properties only and are not intended to form a logically complete set; (ii) is immediately justified and so are (O) and ($*$). However, (†) and (‡) are derived from the discussion of some relation-algebraic results, when $\|$ is replaced in terms of heterogeneous relational algebra, see section 4; they are also justified from interleaving models for process algebra. ◇

Since a relation algebra with parallelism operator is a complete atomic boolean algebra with operators, we are able to establish the concept of *atom structures* (cf. [12] section 3 or [9] 2.7.32ff.). This concept uses an appropriate analogon of the well-known correspondence between fields of sets and complete atomic boolean algebras. It is essential for combinatorial model construction for boolean algebras with operators, in particular for relation algebras [12, 14, 16, 7]. We will make use of it for the construction of the action graph model, which will yield the semantics of our process-algebraic language.

2.3 Definition. A structure $\mathfrak{A} = (A, U, C, T, \P)$ is called a **relational atom structure with parallelism relation** iff the following conditions hold:

(i) U is a subset of A which is called the set of **units**;

(ii) C is a ternary relation on A, i.e. $C \subseteq A \times A \times A$, the so-called **incidence relation**;

(iii) T is a function $A \to A$ which is called **pre-transposition**, where $T \circ T = id_A$ holds;

(iv) $\P$ is a ternary relation on A, i.e. $\P \subseteq A \times A \times A$, the so-called **parallelism relation**;

(v) the **(right) pre-identity rule**:

$$\forall a \in A\colon\ \exists u \in U\colon\ C[a,a,u], \quad \forall a, a' \in A,\ u \in U\colon\ C[a',a,u] \ \Rightarrow\ a = a';$$

(vi) the **pre-associativity rules**:

$$\forall a,b,c,d,e \in A\colon\ C[d,a,b] \wedge C[e,d,c] \ \Rightarrow\ \exists f \in A\colon C[e,a,f] \wedge C[f,b,c],$$

$$\forall a,b,c,e \in A\colon\ (\exists d \in A\colon \P[d,a,b] \wedge \P[e,d,c]) \ \Leftrightarrow\ (\exists f \in A\colon \P[e,a,f] \wedge \P[f,b,c]);$$

(vii) the **pre-Schröder rule**:

$$\forall a,b,c \in A\colon\ C[c,a,b] \ \Rightarrow\ C[a,c,T(b)] \wedge C[b,T(a),c];$$

($viii$) the **pre-automorphicity rule**:

$$\forall a,b,c \in A\colon\ \P[c,a,b] \ \Leftrightarrow\ \P[T(c),T(a),T(b)];$$

(ix) the **unsharpness rules**:

$$\exists a,b,c,d,e \in A\colon\ (\exists a',b' \in A\colon C[e,a',b'] \wedge \P[a',a,b] \wedge \P[b',c,d]) \ \not\Leftrightarrow$$
$$\not\Leftrightarrow\ (\exists a'',b'' \in A\colon \P[e,a'',b''] \wedge C[a'',a,c] \wedge C[b'',b,d]).$$
$$\forall a,b,c,d,e \in A\colon\ (\exists a',b' \in A\colon C[e,a',b'] \wedge \P[a',a,b] \wedge \P[b',c,d]) \ \Rightarrow$$
$$\Rightarrow\ (\exists a'',b'' \in A\colon \P[e,a'',b''] \wedge C[a'',a,c] \wedge C[b'',b,d]). \qquad \diamond$$

The following theorem establishes the connection between the operations of relational atom structures to those of relation algebras with parallelism relation. It is essential to the construction of the action graph model presented in the next section.

2.4 Correspondence Theorem. (i) For every relation algebra with parallelism operator $\mathfrak{A} = (A, \vee, \wedge, \overline{\ }, \cdot, {}^{\mathsf{T}}, \|)$ the structure $\mathfrak{At}\,\mathfrak{A} = (At\mathfrak{A}, U, C, {}^{\mathsf{T}}, \P)$ is a relational atom structure with parallelism relation, where

$At\mathfrak{A}$ denotes the set of the atoms of $\mathfrak{A}$, and $U := \{a \in At\mathfrak{A} \mid a \subset I\}$,

$\forall a,b,c \in At\mathfrak{A}\colon\ C[c,a,b] :\Leftrightarrow c \subset a \cdot b, \P[c,a,b] :\Leftrightarrow c \subset a\|b$ hold.

(*ii*) Conversely, to every relational atom structure with parallelism relation $\mathfrak{A} = (A, U, C, T, \P)$ a relation algebra with parallelism operator $\mathfrak{Cm}\,\mathfrak{A} = (\mathbb{P}(A), \cup, \cap, \overline{\ }, \cdot, {}^{\mathsf{T}}, \|)$, called the **complex algebra of** $\mathfrak{A}$, is associated, where for all $\mathcal{S}, \mathcal{T} \subseteq A$:

$\mathcal{S} \cdot \mathcal{T} := \{c \in A \mid \exists a \in \mathcal{S}, b \in \mathcal{T}\colon C[c, a, b]\}$, $\mathcal{S}^{\mathsf{T}} := \{T(a) \mid a \in \mathcal{S}\}$,

$\mathcal{S} \| \mathcal{T} := \{c \in A \mid \exists a \in \mathcal{S}, b \in \mathcal{T}\colon \P[c, a, b]\}$, $\mathsf{O} = \emptyset$, $\mathsf{L} = A$, and $\mathrm{I} = U$.

Proof. Since relation algebras with parallelism operators are boolean algebras with operators, it can be derived from literature [12, 9, 16]. In particular, 2.3(*v*) and the first part of 2.3(*vi*) correspond to the second part of 2.1(*i*), 2.3(*vii*) corresponds to (S) of 2.1, the second part of 2.3(*vi*) and 2.3(*viii*) correspond to 2.1(*ii*), and 2.3(*ix*) corresponds to both of (†) and (‡) of 2.1. □

3. The Action Graph Model

To provide our process-algebraic language given by the structure of relation algebra with parallelism operator with an appropriate semantics, the *action graph model* is constructed as an element of the algebraic class defined by 2.1. It follows by theorem 2.4(*ii*) of the preceding section that we are able to give this model in terms of its atom structure. Thus, we first have to determine the set of atoms, and then we specify the corresponding operations.

Atoms are here intended to represent single behaviours of processes. To model such a behaviour we use so-called action graphs. The definition of an action graph is similar to those of labelled partial orders [21] and of action structures (cf. [2] section 2): nodes are labelled by actions, while edges indicate causal dependencies. The decisive difference is the undo mapping, which is motivated by the following imagination: when a user selects the "undo" function, the system is urged to take back the last action carried out by executing the counterpart of this action; thus, for the system there exists a mapping that maps each action to its counterpart.

3.1 Definition. A structure $\mathsf{G} = (\mathcal{E}, \mathcal{A}, .\dagger, \leq, \lambda)$ is called an **action graph** iff the following conditions hold:

(*i*) $\mathcal{E}$ is a *finite* set of **events**;

(*ii*) $\mathcal{A}$ is a non-empty set of **(positive) actions**;

(*iii*) $\dagger$ is a function $\mathcal{A} \cup \mathcal{A}\dagger \to \mathcal{A} \cup \mathcal{A}\dagger$ such that $\forall a \in \mathcal{A}\colon a\dagger \in \mathcal{A}\dagger \wedge a\dagger\dagger = a$ holds and $\mathcal{A}\dagger$ is a set disjoint with $\mathcal{A}$; $\dagger$ is called the **undo mapping**;

(*iv*) $\leq \subseteq \mathcal{E} \times \mathcal{E}$ is a partial order, called the **causal dependence**;

(*v*) λ is a function $\mathcal{E} \to \mathcal{A} \cup \mathcal{A}\dagger$, called the **labelling**.

Let AG denote the set of all *abstract* action graphs, i.e. the set of all action graphs modulo isomorphism: we identify two action graphs if their causal dependence relations are identical upto an isomorphism and their labelling functions are equal upto codomain extensions. ◇

Step by step, we define operations on action graphs to establish the desired relational atom structure with parallelism relation.

a) Basic Operations

3.2 Definition: Action Graph Sequential Composition. Let $\mathsf{G} = (\mathcal{E}, \mathcal{A}, .\dagger, \leq, \lambda)$, $\mathsf{G}' = (\mathcal{E}', \mathcal{A}', .\dagger', \leq', \lambda')$ be two action graphs. Then we define the action graph $\mathsf{G}'' := \mathsf{G} \circ \mathsf{G}'$ by

$$\begin{array}{l} \mathcal{E}'' := \mathcal{E} + \mathcal{E}'; \quad \mathcal{A}'' := \mathcal{A} \cup \mathcal{A}'; \quad \dagger'' := \dagger + \dagger'; \\ \leq'' := \{\, (in\mathcal{E}(e), in\mathcal{E}(e')) \,|\, e, e' \in \mathcal{E} \;\wedge\; e \leq e' \,\} \cup \\ \qquad \cup \{\, (in\mathcal{E}(e), in\mathcal{E}'(e')) \,|\, e \in \mathcal{E}, e' \in \mathcal{E}' \,\} \cup \{\, (in\mathcal{E}'(e), in\mathcal{E}'(e')) \,|\, e, e' \in \mathcal{E}' \;\wedge\; e \leq' e' \,\}; \\ \lambda'' := \lambda + \lambda', \end{array}$$

where $\mathcal{E} + \mathcal{E}'$, $in\mathcal{E}$, $in\mathcal{E}'$, $\dagger + \dagger'$, and $\lambda + \lambda'$ concern notations when using direct sums. $\diamond$

3.3 Definition: Action Graph Transposition. Let $\mathsf{G} = (\mathcal{E}, \mathcal{A}, .\dagger, \leq, \lambda)$ be an action graph. Then we define the action graph $\mathsf{G}' := \mathsf{G}^{\mathsf{T}}$ by

$$\mathcal{E}' := \mathcal{E}; \quad \mathcal{A}' := \mathcal{A}; \quad \dagger' := \dagger; \quad \leq' := \geq; \quad \lambda' := \lambda \circ \dagger,$$

i.e. $\leq'$ is the converse ordering, and $\forall e \in \mathcal{E}\colon\ \lambda'(e) = \lambda(e)\dagger$ holds. $\diamond$

3.4 Definition: Action Graph Sum. Let $\mathsf{G} = (\mathcal{E}, \mathcal{A}, .\dagger, \leq, \lambda)$, $\mathsf{G}' = (\mathcal{E}', \mathcal{A}', .\dagger', \leq', \lambda')$ be two action graphs. Then we define the action graph $\mathsf{G}'' := \mathsf{G} + \mathsf{G}'$ by

$$\mathcal{E}'' := \mathcal{E} + \mathcal{E}'; \quad \mathcal{A}'' := \mathcal{A} \cup \mathcal{A}'; \quad \dagger'' := \dagger + \dagger'; \quad \leq'' := in\mathcal{E}(\leq) \cup in\mathcal{E}'(\leq'); \quad \lambda'' := \lambda + \lambda'. \quad \diamond$$

3.5 Definition: Empty Action Graph. We define the action graph $\Lambda = (\mathcal{E}, \mathcal{A}, .\dagger, \leq, \lambda)$ as follows:

$\mathcal{E} := \emptyset$; let $\mathcal{A}$ be any arbitrary set and let $\dagger$ be a corresponding undo mapping; $\leq := \emptyset$; let λ be the unique function $\emptyset \to \mathcal{A} \cup \mathcal{A}\dagger$. $\diamond$

b) Parallel Composition

To ensure (‡) of 2.1 we do not define parallel composition as sum of graphs (like e.g. in [21]), but we use the notion of a *partial sequentialisation* of an action graph (cf. [2] 2.1, and cf. also the notions of *augment* and *augment closure* [21, 11]).

3.6 Definition: Action Graph Partial Sequentialisation. Let $\mathsf{G} = (\mathcal{E}, \mathcal{A}, \leq, \lambda)$, $\mathsf{G}' = (\mathcal{E}', \mathcal{A}', \leq', \lambda')$ be two action graphs. G' is called a **partial sequentialisation** of G iff $\leq \subseteq \leq'$. Then, let $\sigma(\mathsf{G})$ denote the set of all partial sequentialisations of G. $\diamond$

3.7 Definition: Action Graph Parallel Composition. Let $\mathsf{G} = (\mathcal{E}, \mathcal{A}, \leq, \lambda)$, $\mathsf{G}' = (\mathcal{E}', \mathcal{A}', \leq', \lambda')$ be two action graphs. Then we define the set of action graphs $\mathcal{G}'' := \mathsf{G} \| \mathsf{G}'$ by $\mathcal{G}'' = \sigma(\mathsf{G} + \mathsf{G}')$. $\diamond$

c) Multiplication

Multiplication or composition of relations is usual intended to represent the control principle of sequential composition. However, in order to achieve a modelling of "undo" this view is only the starting point for the incidence relation on action graphs, since (S) of 2.1 and 2.3(*vii*), respectively, have to be satisfied.

3.8 Action Graph Incidence Relation. Let G_1, G_2, and G_3 be three action graphs. Then we define $\mathsf{C} \subseteq AG^3$ to be the smallest relation such that the following conditions hold:

(*i*) $\mathsf{G}_1 = \mathsf{G}_2 \circ \mathsf{G}_3 \implies \mathsf{C}[\mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_3]$;

(*ii*) $\mathsf{C}[\mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_3] \implies \mathsf{C}[\mathsf{G}_2, \mathsf{G}_1, \mathsf{G}_3^{\mathsf{T}}] \wedge \mathsf{C}[\mathsf{G}_3, \mathsf{G}_2^{\mathsf{T}}, \mathsf{G}_1]$. ◇

3.9 Remark. The unique existence of C can be derived from literature (cf. [14, 16] and also [7] 2.5.17–2.5.19). ◇

d) Results

Since the necessary operations on action graphs are given, we develop, as announced, the action graph model as a complex algebra of the atom structure of action graphs.

3.10 Theorem. $\mathfrak{AG} = (AG, U, \mathsf{C}, {}^{\mathsf{T}}, \P)$ is a relational atom structure with parallelism relation, where

$$U := \{\Lambda\},\ \mathsf{C} \text{ is known from 3.8, } {}^{\mathsf{T}} \text{ from 3.3, and } \forall \mathsf{G}, \mathsf{G}', \mathsf{G}'' \in AG\colon\ \P[\mathsf{G}'', \mathsf{G}, \mathsf{G}'] :\Leftrightarrow \mathsf{G}'' \in \mathsf{G} \| \mathsf{G}'.$$

Proof. We have to check all conditions of 2.3: (*i*)–(*iv*) by definition. ad (*v*)–(*viii*): follow from the definitions of action graph sequential/parallel composition, action graph transposition, action graph incidence relation, and empty action graph Λ; we omit details here.

ad (*ix*): The first part can be reduced to

$$\exists\, \mathsf{G}_1,\ \mathsf{G}_2,\ \mathsf{G}_3,\ \mathsf{G}_4 \in AG\colon\ (\mathsf{G}_1 + \mathsf{G}_2) \circ (\mathsf{G}_3 + \mathsf{G}_4) \neq (\mathsf{G}_1 \circ \mathsf{G}_3) + (\mathsf{G}_2 \circ \mathsf{G}_4).$$

Choose for all G_i the same one-event action graph $\mathbb{1} := (\{*\}, \{!\}, =, * \mapsto !)$, then we get the situation depicted in FIGURE 1.

For the second part, it suffices to show that

$$\begin{aligned} &\forall \mathsf{G}, \dot{\mathsf{G}}, \ddot{\mathsf{G}} \in AG\colon\ \mathsf{C}[\mathsf{G}, \dot{\mathsf{G}}, \ddot{\mathsf{G}}] \wedge \dot{\mathsf{G}} \in \sigma(\mathsf{G}_1 + \mathsf{G}_2) \wedge \ddot{\mathsf{G}} \in \sigma(\mathsf{G}_3 + \mathsf{G}_4) \implies \\ &\quad \implies \exists \mathsf{G}', \mathsf{G}'' \in AG\colon\ \mathsf{C}[\mathsf{G}', \mathsf{G}_1, \mathsf{G}_3] \wedge \mathsf{C}[\mathsf{G}'', \mathsf{G}_2, \mathsf{G}_4] \wedge \mathsf{G} \in \sigma(\mathsf{G}' + \mathsf{G}'') \end{aligned}$$

holds. This is obviously satisfied when replacing C by the relationship in 3.8(*i*), i.e. action graph sequential composition is considered. The problem arising from 3.8(*ii*) may be tackled by taking the pre-automorphicity rule 2.3(*viii*) into account. □

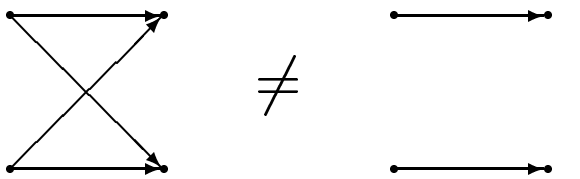

Figure 1: Situation for the satisfaction of the **unsharpness rule**.

3.11 Corollary. The structure $\mathfrak{P}_{AG} = \mathfrak{Cm}\,\mathfrak{AG}$, i.e. the complex algebra of $\mathfrak{AG}$ (see 2.4(*ii*)), is a relation algebra with parallelism operator.

Proof. Immediately by 2.4(*ii*) and 3.10. □

3.12 Remark. From the definition of a complex algebra it is clear that set union $\cup$ plays the rôle of *nondeterministic choice*, for it builds up each a variety of possible behaviours (cf. e.g. the notion of *process* in [21] 2.3). ◇

The second result concerns the modelling of "undo"; it is investigated to what extent transposition together with the *Schröder equivalences*, (S) of 2.1, achieves an appropriate modelling provided that transposition is considered in the action graph model $\mathfrak{P}_{AG}$.

3.13 Theorem. Let $\mathsf{G}, \mathsf{G}', \mathsf{G}'' \in AG$ be three action graphs. Then the $\mathfrak{P}_{AG}$-expression $\{\mathsf{G}'\} \cdot \{\mathsf{G}\} \cdot \{\mathsf{G}^\mathsf{T}\} \cdot \{\mathsf{G}''\}$ includes the action graphs $\mathsf{G}' \circ \mathsf{G} \circ \mathsf{G}^\mathsf{T} \circ \mathsf{G}''$ as well as $\mathsf{G}' \circ \mathsf{G}''$.

Proof. First, it is clear that $\circ$ (see 3.2) and $\cdot$ (see 2.4(*ii*)) are both associative. Second, it is easy to see by (S) of 2.1 that the empty action graph Λ is contained by every product of the form $R \cdot R^\mathsf{T}$, even if $R = \{\mathsf{G}\}$ holds. The rest is concluded from the evident monotonicity of multiplication $\cdot$. □

3.14 Remark. The preceding theorem confirms the intuition that "undo" appears as the non-deterministic possibility to omit the behaviour to be undone and its immediately following abandonment by the contrary happening from the beginning. Instead of this theorem one would rather like to prove the analogous result when replacing $\circ$ by a more flexible connection operator; note that in $\mathsf{G} \circ \mathsf{G}'$ all events of G' are greater than any event of G wrt. causal dependence relation. Such a flexible connection operator may be stated as $\circ_\mathcal{I}$ where for $\mathsf{G} \circ_\mathcal{I} \mathsf{G}'$ the index $\mathcal{I}$ is a set of pairs of events such that the first component of each pair is a maximum wrt. $\leq_\mathsf{G}$ while the second component is a minimum wrt. $\leq_{\mathsf{G}'}$. It only remains to state $\mathsf{C}_\mathcal{I}$ and $\cdot_\mathcal{I}$, resp., and to require the pre-Schröder rule, the Schröder equivalences, resp., and the corresponding associativity rules between different connection operations. ◇

4. A Result on Heterogeneous Abstract Relation Algebras

Heterogeneous abstract relational algebra is the algebraisation of the calculus of relations whose domain and codomain are not isomorphic. In this section, we consider those relation-algebraic results leading to our axiomatisation of the parallelism operator, and we investigate the question whether a relation algebra with parallelism operator can be embedded in a heterogeneous abstract relation algebra by expressing the parallelism operation in terms of relational algebra. Namely, this would lead to a construction of an example of a *non-representable* heterogeneous relation algebra from the action graph model $\mathfrak{P}_{AG}$, i.e. an example with a more intuitive background in contrast to the generally less intuitive descriptions by their atom structures (for the topic of heterogeneous relational atom structures see [7] 5.1). However, we can show that this construction fails.

The following definition is due to [25] A.2.1 and [7] 4.1.1, respectively.

4.1 Definition. A structure $\mathfrak{H} = (H, \vee, \wedge, \overline{\ }, \cdot, {}^{\mathsf{T}})$ is called a **heterogeneous abstract relation algebra** iff the following conditions hold:

(i) For each $R \in H$ the following terms are defined: $\overline{R}, R^{\mathsf{T}}, R^{\mathsf{T}}R, RR^{\mathsf{T}}$, and $R \vee R$.

(ii) For each $R \in H$ define $\mathfrak{H}_{\vee,R} := \{S \in H \mid R \vee S \text{ is defined}\}$, the so-called **component of $\mathfrak{H}$ relative to** R. Every $\mathfrak{H}_{\vee,R}$ forms a complete atomic boolean algebra.
If a component is closed under multiplication, then we will call it a **homogeneous** one. Moreover, the following holds for arbitrary components:

$$\forall R \in H\colon QR \text{ is defined} \Longrightarrow \forall S \in \mathfrak{H}_{\vee,R} : QS \text{ is defined.}$$

(iii) For all $Q, R, S \in H$, if one of $Q(RS)$, $(QR)S$ is defined, then, both are defined, and $Q(RS) = (QR)S$ holds.

(iv) For every component $\mathfrak{H}_{\vee,R}$ there is $\mathrm{I}_R \in H$ such that $\forall S \in \mathfrak{H}_{\vee,R}\colon \mathrm{I}_R S = S$.

(v) The **complement-free Schröder equivalences** hold: For all $Q, R, S \in H$, if one of $QR \wedge S$, $Q \wedge SR^{\mathsf{T}}$, and $R \wedge Q^{\mathsf{T}}S$ is defined, then, all of them are defined, and

$$QR \wedge S = \mathsf{0} \iff Q \wedge SR^{\mathsf{T}} = \mathsf{0} \iff R \wedge Q^{\mathsf{T}}S = \mathsf{0}$$

holds.

A heterogeneous abstract relation algebra $\mathfrak{H}$ is called a **homogeneous abstract relation algebra** iff the universe H itself is a homogeneous component of $\mathfrak{H}$. ◇

4.2 Definition. A heterogeneous abstract relation algebra $\mathfrak{H}$ is called **representable** iff $\mathfrak{H}$ is isomorphic to a heterogeneous abstract relation algebra $\mathfrak{H}' = (H', \cup, \cap, \overline{\ }, \circ, {}^{-1})$, where H' is a set of relations and the operations are the "usual" operations on relations. ◇

In order to give an interpretation of the parallel composition operator $\|$ in terms of relational algebra so that (†) and (‡) of 2.1 may be expressed, we use the notion of a relational product. A relational product is a pair (π, ρ) of elements of a heterogeneous abstract relation algebra satisfying certain conditions so that π and ρ imitate the behaviour of a pair of projections belonging to a direct product in the category of sets.

4.3 Definition. Let $\mathfrak{H} = (H, \vee, \wedge, \overline{\ }, \cdot, ^{\mathsf{T}})$ be a heterogeneous abstract relation algebra. $(\pi, \rho) \in H \times H$ is called a **(binary) relational product** iff the following conditions are satisfied:

$$(i) \quad \pi^{\mathsf{T}}\pi = \mathrm{I},\ \rho^{\mathsf{T}}\rho = \mathrm{I}; \qquad (ii) \quad \pi\pi^{\mathsf{T}} \wedge \rho\rho^{\mathsf{T}} = \mathrm{I}; \qquad (iii) \quad \pi^{\mathsf{T}}\rho = \mathsf{L}. \qquad \diamond$$

4.4 Definition. Let $\mathfrak{H}$ be a heterogeneous abstract relation algebra, and let (π, ρ) be a relational product in $\mathfrak{H}$. Then we define

$$\forall R, S \in H \text{ appropriately: } R\|_{(\pi,\rho)}S := \pi R\pi^{\mathsf{T}} \wedge \rho S\rho^{\mathsf{T}}. \qquad \diamond$$

By some of the following results it turns out that a relation-algebraic model of (†) is really heterogeneous (4.6) and non-representable (4.5). The decisive result is the result of [28] (4.7), which allows the conclusion that we are not able to construct such a model from the action graph model $\mathfrak{P}_{AG}$ (4.8) when replacing $\|$ by the construct of 4.4.

4.5 Proposition. Let $\mathfrak{H}$ be a heterogeneous abstract relation algebra, and let (π, ρ) be a relational product in $\mathfrak{H}$. Then the following holds:

$$(\ddagger) \quad \forall Q,\ R,\ S,\ T \in H \text{ appropriately: } (Q\|_{(\pi,\rho)}R)(S\|_{(\pi,\rho)}T) \subset (QS)\|_{(\pi,\rho)}(RT).$$

If $\mathfrak{H}$ is representable, then also the equality holds:

$$(\ddagger') \quad \forall Q,\ R,\ S,\ T \in H \text{ appropriately: } (Q\|_{(\pi,\rho)}R)(S\|_{(\pi,\rho)}T) = (QS)\|_{(\pi,\rho)}(RT).$$

Proof. (‡) is obviously obtainable from the theory of relation algebras. (‡′) follows either directly by [13] Theorem 4.7(i) or as a corollary of [28] Lemma 2.6.2(i). If restricted to homogeneous abstract relation algebras, see also [26] 4.1(viii) (i.e. the proof of Maddux). □

4.6 Proposition. Any *homogeneous* abstract relation algebra with a relational product is representable, i.e. (‡′) of 4.5 holds.

Proof. See [15] Corollary 8 or [26] 8.4(iii). For the algebraic version [15] the decisive conditions are 4.6(i),(iii): $\pi^{\mathsf{T}}\pi \subset \mathrm{I}$, $\rho^{\mathsf{T}}\rho \subset \mathrm{I}$, and $\pi^{\mathsf{T}}\rho = \mathsf{L}$. □

4.7 Theorem *(due to [28] 2.6.2(i))*. Let $\mathfrak{H} = (H, \vee, \wedge, \overline{\ }, \cdot, ^{\mathsf{T}})$ be a heterogeneous abstract relation algebra with a relational product (π, ρ). Further let $Q, R, S, T \in H$ such that $Q\|_{(\pi,\rho)}R$ and $S\|_{(\pi,\rho)}T$ are defined. If there exists two further appropriate relational products, then (‡′) of 4.5 holds for $\mathfrak{H}$.

Proof. The two announced relational products are pairs (σ, τ) and (χ, υ) such that

$$S\|_{(\chi,\upsilon)}(T\|_{(\sigma,\tau)}(Q\|_{(\pi,\rho)}R))$$

is defined. If this requirement is satisfied, then we can define the two elements X, Y of $\mathfrak{H}$ like in Zierer's proof as follows:

$$X := \tau\pi Q\pi^{\mathsf{T}} \wedge (\sigma \wedge \tau\rho R)\rho^{\mathsf{T}}, \quad Y := (\chi \wedge \upsilon\tau\pi Q)\pi^{\mathsf{T}} \wedge \upsilon(\sigma \wedge \tau\rho R)\rho^{\mathsf{T}}.$$

One also obtains: $\tau^{\mathsf{T}}X = Q\|_{(\pi,\rho)}R$, $\upsilon^{\mathsf{T}}Y = X$, and, with exactly the same estimation as in Zierer's proof, that $Y^{\mathsf{T}}Y \subset \mathrm{I}$ holds. The rest of the proof works like in [28] on p. 28 starting with $(Q\|_{(\pi,\rho)}R)(S\|_{(\pi,\rho)}T) = \tau^{\mathsf{T}}\upsilon^{\mathsf{T}}Y(S\|_{(\pi,\rho)}T)$. □

4.8 Corollary. There is no heterogeneous abstract relation algebra $\mathfrak{H}_{AG}$ that the relation algebra with parallelism operator $\mathfrak{P}_{AG}$ is embedded into such that the carrier set $\mathbb{P}(AG)$ appears as a (homogeneous) component of $\mathfrak{H}_{AG}$ and parallel composition may be replaced as follows: For all $\mathcal{R}, \mathcal{S} \subseteq AG$ there exists a relational product $(\pi_{\mathcal{R},\mathcal{S}}, \rho_{\mathcal{R},\mathcal{S}})$ such that $\mathcal{R}\|\mathcal{S} = \mathcal{R}\|_{(\pi_{\mathcal{R},\mathcal{S}},\rho_{\mathcal{R},\mathcal{S}})}\mathcal{S}$ is consistently satisfiable.

Proof. Obvious from the proof of the preceding theorem: at least, $\mathfrak{H}_{AG}$ would have to contain enough relational products so that any quaternary parallel composition may be formed. □

5. Conclusion

In this paper we have left the mainstream for four times: First, a process-algebraic language is specified by the mathematical method of universal algebra: its syntax is given by the definition of the corresponding algebraic class, while its semantics is a single member of that class; thus, the usual separation of syntax and semantics has been avoided.

Second, since we are interested in a (non-complete) algebraic characterisation of parallel composition, we have accepted the inequality (†) as an axiom for technical convenience; alternatively, we could have defined the class of relation algebras with parallelism operator by omitting (†) and we could have checked the validity of (†) for the selected models. This inequality is justified by the usual models of process algebra and by some models of temporal logic (cf. [19] Theorem 2.13). However, whether (†) in the proposed relation-algebraic style may be proved to hold in heterogeneous abstract relation algebras, is still an open question. Nevertheless, it has turned out by results of the preceding section that replacing the parallelism operator by the proposed relation-algebraic construct leads to the invalidity of (†). The negation of (†), which is an equation, has practical relevance, too. It appears as an essential component of the verification concept of *communication closed layers* [4, 11], where $Q\|R$ and $S\|T$ of (†) are thought to be *layers* of the program $(QS)\|(RT)$; to achieve this, for the multiplication in $(Q\|R)\cdot(S\|T)$ an additional condition is required (cf. the notion of *safe decomposition* [4] and that of *conflict-free* composition [11]). Thus, we have found an appropriate modelling of the case of communication-closed layers. Another aspect of the relation-algebraic construct $\|_{(\pi,\rho)}$ is the possible interpretation of it as cartesian product operation of relations modulo an appropriate tuple permutation; in this sense it would rather correspond to Pratt's *orthocurrence* operator [21].

Third, we stress the fact that we do not use relational algebra aiming at concrete binary relations, we rather use the fact that relation algebras are boolean algebras with operators and they have associated atom structures [12, 14, 16]; adding a concurrency operator to the framework of binary relations has been done in former years (see [20]). Thus, we do not use graphs as relations by their associated relation, we accept graphs as atomic elements of an abstract relation algebra.

Finally, one of the decisive objectives of the paper has been the modelling of "undo" by the transposition operator, which arises from relational algebra as an operator additional with respect to usual process algebra. In Pratt's work (cf. [21, 22] among others) transposition appears as the causality-reversal operation, which is not used for process specification in practice. Our transposition is indeed also a causality-reversal operation, but the effect of taking back actions is taken into account: each action is mapped to its counterpart by the undo mapping, like a process would carry out the "undo" operation.

Acknowledgements

I wish to thank Manfred Broy for fruitful discussions about this topic [8]. I am also indebted to Vaughan R. Pratt for a valuable hint on the topic and to Tobias Nipkow, Rudolf Berghammer, and Rainer Weber for critical remarks on a draft of this paper.

References

[1] Bergstra, J.A., Klop, J.W., *Process algebra: Specification and verification in bisimulation semantics*, *in:* Hazewinkel, M., Lenstra, J.K., Meertens, L.G.L.T. (eds.), *Mathematics and Computer Science II*, CWI Monographs **4** (1986) 61–94, North-Holland Publ.Co.

[2] Broy, M., *Operational and denotational semantics with explicit concurrency*, *to appear in:* Fundamenta Informaticae **XV** (1992)

[3] Cardoso, R., *Untersuchung paralleler Programme mit relationenalgebraischen Methoden*, Technische Universität München, Diploma Thesis (1982)

[4] Elrad, Tz., Francez, N., *Decomposition of distributed programs into communication closed layers*, *in:* Science of Computer Programming **2** (1982) 155–173, North-Holland Publ.Co.

[5] van Glabbeek, R.J., *The linear time – branching time spectrum*, *in:* Baeten, J.C.M., Klop, J.W. (eds.), *CONCUR '90*, Lecture Notes in Computer Science **458** (1990) 278–297, Springer-Verlag

[6] Goguen, J.A., *On homomorphisms, correctness, termination, unfoldments and equivalence of flow diagram programs*, *in:* Journal of Computer and System Sciences **8** (1974) 333–365, Academic Press Inc.

[7] Gritzner, T.F., *Die Axiomatik abstrakter Relationenalgebren: Darstellung der Grundlagen und Anwendung auf das Unschärfeproblem relationaler Produkte*, Technische Universität München, Internal Report TUM-INFO-04-91-I00 (1991)

[8] Gritzner, T.F., Broy, M., *Parallelism algebras: a link between process algebras and abstract relation algebras?*, Technische Universität München, Technical Report, SFB-Bericht Nr. 342/15/91 A (1991)

[9] Henkin, L., Monk, J.D., Tarski, A., **Cylindric Algebras, Parts I & II**, *series:* Studies in Logic and the Foundations of Mathematics **64** (1971) & **115** (1985), North-Holland Publ.Co.

[10] Hoare, C.A.R., He, Jifeng, *The weakest prespecification, parts I&II*, *in:* Fundamenta Informaticae **IX** (1986) 51–84 & 217–252, North-Holland Publ.Co. (in association with the Polish Mathematical Society)

[11] Janssen, W., Poel, M., Zwiers, J., *Action systems and action refinement in the development of parallel systems*, *in:* Baeten, J.C.M, Groote, J.F. (eds.), *CONCUR '91*, Lecture Notes in Computer Science **527** (1991) 298–316, Springer-Verlag

[12] Jónsson, B., Tarski, A., *Boolean algebras with operators, parts I&II*, *in:* American Journal of Mathematics **73** (1951) 891–939 & **74** (1952) 127–167, The John Hopkins University Press

[13] Leischner, M., Gritzner, T.F., *Relating relational products to categorical products*, Universität München, Internal Report No. 9201 (1992)

[14] Lyndon, R.C., *The representation of relation(al) algebras, parts I&II*, *in:* Annals of Mathematics (II) **51** (1950) 707–729 & **63** (1956) 294–307, Princeton University Press

[15] Maddux, R., *Some sufficient conditions for the representability of relation algebras*, *in:* Algebra Universalis **8** (1978) 162–172, Birkhäuser Verlag

[16] Maddux, R., *Finite integral relation algebras*, *in:* Comer, S.D. (ed.), *Universal Algebra and Lattice Theory*, Lecture Notes in Mathematics **1149** (1985) 175–197, Springer-Verlag

[17] Olderog, E.-R., *Semantics of concurrent processes: the search for structure and abstraction, parts I&II*, *in:* Bulletin of the EATCS **28** (1986) 73–97 & **29** (1986) 96–117

[18] Olderog, E.-R., Hoare, C.A.R., *Specification-oriented semantics for communicating processes*, *in:* Acta Informatica **23** (1986) 9–66, Springer-Verlag

[19] Paech, B., *Concurrency as a Modality*, Universität München, Doctoral Thesis, *available as* Internal Report No. 91/01 (1991)

[20] Peleg, D., *Concurrent Dynamic Logic*, *in:* Journal of the ACM **34:2** (1987) 450–479

[21] Pratt, V., *Modelling concurrency with partial orders*, *in:* International Journal of Parallel Computing **15** (1986) 33–71, Plenum Publ.Co.

[22] Pratt, V., *Event spaces and related structures*, manuscript, Stanford, Jan. 1992 (*available via anonymous ftp from boole.stanford.edu (36.8.0.65)*)

[23] de Roever, W.P., *A formalization of various parameter mechanisms as products of relations within a calculus of recursive program schemes*, *in: Théorie des Algorithmes, des Langages et de la Programmation*, Séminaires d'IRIA (1972) 55–88

[24] Schmidt, G., *Programs as partial graphs I: flow equivalence and correctness*, *in:* Theoretical Computer Science **15** (1981) 1–25, Elsevier Science Publishers B.V. (North-Holland)

[25] Schmidt, G., Ströhlein, Th., **Relations and Graphs**, Springer-Verlag, *in preparation, still available in German*

[26] Tarski, A., Givant, S., **A Formalization of Set Theory Without Variables**, AMS Colloquium Publications **41** (1987)

[27] Winskel, G., *Event structures*, *in:* Brauer, W., Reisig, W., Rosenberg, G. (eds.), *Petri-Nets: Applications and Relationships to Other Models of Concurrency*, Lecture Notes in Computer Science **255** (1987) 325–392, Springer-Verlag

[28] Zierer, H., *Programmierung mit Funktionsobjekten: Konstruktive Erzeugung semantischer Bereiche und Anwendung auf die partielle Auswertung*, Technische Universität München, Doctoral Thesis, *available as* Technical Report TUM–I8803 (1988)

Appendix

The appendix is devoted for the proofs left up in the paper for convenience. Note that only proofs of the paper's concern are presented, while others are assumed to be easily derived from literature.

A. Proof of 2.4

Ad (i):

Here we only have to cope with the parallelism relation ¶ defined by $\P[c,a,b] :\Leftrightarrow c \subset a\|b$.

Ad 2.3(vi), part II: By associativity of $\|$ we get:

$$\forall a,b,c,e\colon\ e \subset (a\|b)\|c \iff e \subset a\|(b\|c).$$

By completeness of $\mathfrak{A}$ and (O), $(*)$ of 2.1 $\left((a\|b)\|c = \bigvee_{d\subset a\|b} d\|c \text{ etc.}\right)$ we arrive at

$$\forall a,b,c,e\colon\ (\exists d\colon d \subset a\|b \wedge e \subset d\|c) \iff (\exists f\colon e \subset a\|f \wedge f \subset b\|c).$$

Ad 2.3(viii): Let $c \subset a\|b$ be given. Clearly, $c^{\mathsf{T}} \subset (a\|b)^{\mathsf{T}}$ holds (simple relational algebra); this formula is even equivalent to the given one. It is also, in turn, equivalent to $c^{\mathsf{T}} \subset a^{\mathsf{T}}\|b^{\mathsf{T}}$ by 2.1(ii).

Ad 2.3(ix): Repeat the proof steps belonging to the pre-associativity rule 2.3(vi).

Ad (ii):

We only deal with $\|$ defined by $\mathcal{A}\|\mathcal{B} := \{c \in A \mid \exists a\in\mathcal{A}, b\in\mathcal{B}\colon\ \P[c,a,b]\}$.

Ad 2.1(ii), part I: The associativity of $\|$ is immediately established by 2.3(vi), part II.

Ad 2.1(ii), part II: By 2.3$(viii)$ we have

$$\begin{aligned}(\mathcal{A}\|\mathcal{B})^{\mathsf{T}} &= \{c \mid \exists a,b\colon\ \P[c,a,b]\}^{\mathsf{T}} = \{T(c) \mid \exists a,b\colon\ \P[c,a,b]\}\\ &= \{T(c) \mid \exists a,b\colon\ \P[T(c),T(a),T(b)]\} = \{d \mid \exists a,b\colon\ \P[d,T(a),T(b)]\}\\ &= \{T(a) \mid a\in\mathcal{A}\}\|\{T(b) \mid b\in\mathcal{B}\} = \mathcal{A}^{\mathsf{T}}\|\mathcal{B}^{\mathsf{T}}.\end{aligned}$$

Ad 2.1(iii): (O) and $(*)$ are immediately clear from the definition of $\|$, while the unsharpness rules 2.3(ix) establish (†) and (‡).

B. Proof of 3.9

Proposition. C of 3.8 exists and is uniquely determined.

Proof. Due to literature [14, 16]: Put

$$\begin{aligned}[\mathsf{G}_1,\mathsf{G}_2,\mathsf{G}_3] \;:=\; &\{(\mathsf{G}_1,\mathsf{G}_2,\mathsf{G}_3),(\mathsf{G}_2,\mathsf{G}_1,\mathsf{G}_3^{\mathsf{T}}),(\mathsf{G}_3,\mathsf{G}_2^{\mathsf{T}},\mathsf{G}_1),\\ &(\mathsf{G}_1^{\mathsf{T}},\mathsf{G}_3^{\mathsf{T}},\mathsf{G}_2^{\mathsf{T}}),(\mathsf{G}_2^{\mathsf{T}},\mathsf{G}_3,\mathsf{G}_1^{\mathsf{T}}),(\mathsf{G}_3^{\mathsf{T}},\mathsf{G}_1^{\mathsf{T}},\mathsf{G}_2)\},\end{aligned}$$

then $[\mathsf{G}_1,\mathsf{G}_2,\mathsf{G}_3]$ is called the **cycle** of G_1, G_2, and G_3. It is easily verified by application of the rules 3.8(i),(ii) that C is the union of all cycles of the form $[\mathsf{G}\circ\mathsf{G}',\mathsf{G},\mathsf{G}']$: a cycle is closed wrt. the rule 3.8(ii), which corresponds to 2.3(vii). □

C. Completion of the Proof of 3.10

Ad 2.3(ix), part II: Recall in mind that we have had to show

$$\forall \mathsf{G},\dot{\mathsf{G}},\ddot{\mathsf{G}}\in AG\colon\ \mathsf{C}[\mathsf{G},\dot{\mathsf{G}},\ddot{\mathsf{G}}]\wedge\dot{\mathsf{G}}\in\sigma(\mathsf{G}_1+\mathsf{G}_2)\wedge\ddot{\mathsf{G}}\in\sigma(\mathsf{G}_3+\mathsf{G}_4)\implies$$
$$\implies\exists \mathsf{G}',\mathsf{G}''\in AG\colon\ \mathsf{C}[\mathsf{G}',\mathsf{G}_1,\mathsf{G}_3]\wedge\mathsf{C}[\mathsf{G}'',\mathsf{G}_2,\mathsf{G}_4]\wedge\mathsf{G}\in\sigma(\mathsf{G}'+\mathsf{G}'').$$

Lemma 1: If $\mathsf{C}[\mathsf{G}_1,\mathsf{G}_2,\mathsf{G}_3]$ holds, one of the following possibilities applies:

(i) $\mathsf{G}_1=\mathsf{G}_2\circ\mathsf{G}_3$; (ii) $\mathsf{G}_2=\mathsf{G}_1\circ\mathsf{G}_3^\mathsf{T}$; (iii) $\mathsf{G}_3=\mathsf{G}_2^\mathsf{T}\circ\mathsf{G}_1$. □

Convention 1: We write $\mathsf{G}\sqsupseteq\mathsf{G}'$ for $\mathsf{G}\in\sigma(\mathsf{G}')$, i.e. $\leq_\mathsf{G}\supseteq\leq_{\mathsf{G}'}$. Further, according to the definition of $\leq''$ in 3.2 we rewrite $\mathsf{G}''=\mathsf{G}\circ\mathsf{G}'$ as $\mathsf{G}+(\mathsf{G}\to\mathsf{G}')+\mathsf{G}'$. ◇

Convention 2: We assume that we are allowed to use the operations $+$, $\circ$, $\cdot$, and $^\mathsf{T}$ on subsets of action graphs like for $\mathfrak{Cm}\,\mathfrak{AG}$. (Therefore, for $\cdot$ and $^\mathsf{T}$ see 2.4(ii).) ◇

Lemma 2: (i) $\sigma(\mathsf{G}\circ\mathsf{G}')=\sigma(\mathsf{G})\circ\sigma(\mathsf{G}')$, (ii) $\sigma(\mathsf{G}^\mathsf{T})=\sigma(\mathsf{G})^\mathsf{T}$, (iii) $\mathsf{G}\sqsupseteq\mathsf{G}'\implies\sigma(\mathsf{G})\subseteq\sigma(\mathsf{G}')$, and (iv) $\sigma(\mathsf{G})+\sigma(\mathsf{G}')\subseteq\sigma(\mathsf{G}+\mathsf{G}')$ hold.

Proof: Ad (i): "$\supseteq$": obvious. "$\subseteq$" is established by $\mathsf{G}\circ\mathsf{G}'=\mathsf{G}+(\mathsf{G}\to\mathsf{G}')+\mathsf{G}'$ and by the knowledge of $\sigma(\mathsf{G}\to\mathsf{G}')=\{(\mathsf{G}\to\mathsf{G}')\}$. Ad (ii), (iii): obvious. Ad (iv): also obvious. But, note that (cf. conv. 1) $\mathsf{G}\circ\mathsf{G}'\sqsupseteq\mathsf{G}+\mathsf{G}'$ holds, whereas for $\mathsf{G},\mathsf{G}'\neq\Lambda$ the relationship $\mathsf{G}\circ\mathsf{G}'\notin\sigma(\mathsf{G})+\sigma(\mathsf{G}')$ holds. □

Lemma 3: The pre-automorphicity rule 2.3$(viii)$ is equivalent to $(\mathsf{G}+\mathsf{G}')^\mathsf{T}=\mathsf{G}^\mathsf{T}+\mathsf{G}'^\mathsf{T}$. □

Lemma 4: $\mathsf{G}\circ(\mathsf{G}_3+\mathsf{G}_4)\sqsupseteq(\mathsf{G}_1\circ\mathsf{G}_3)+(\mathsf{G}_2\circ\mathsf{G}_4)\implies\mathsf{G}\sqsupseteq\mathsf{G}_1+\mathsf{G}_2$ holds.

Proof: I.e. $\mathsf{G}+\mathsf{G}_3+\mathsf{G}_4+(\mathsf{G}\to\mathsf{G}_3)+(\mathsf{G}\to\mathsf{G}_4)\sqsupseteq\mathsf{G}_1+\mathsf{G}_2+\mathsf{G}_3+\mathsf{G}_4+(\mathsf{G}_1\to\mathsf{G}_3)+(\mathsf{G}_2\to\mathsf{G}_4)$ is assumed to hold. Then, clearly $\mathsf{G}\sqsupseteq\mathsf{G}_1+\mathsf{G}_2$ holds. □

Let $\mathsf{G},\dot{\mathsf{G}},\ddot{\mathsf{G}}\in AG$ be given such that the required properties stated above hold. We deal only with the first two cases caused by l. 1, since the last case (iii) may be proved analogously to (ii).

Case 1: $\mathsf{G}=\dot{\mathsf{G}}\circ\ddot{\mathsf{G}}$:

W.l.o.g. (see l. 2) assume that $\dot{\mathsf{G}}=\mathsf{G}_1+\mathsf{G}_2$ and $\ddot{\mathsf{G}}=\mathsf{G}_3+\mathsf{G}_4$ are satisfied. We claim that $\mathsf{G}'=\mathsf{G}_1\circ\mathsf{G}_3$ and $\mathsf{G}''=\mathsf{G}_2\circ\mathsf{G}_4$ are the desired action graphs:

$$\begin{aligned}\mathsf{G}\ &=\ \mathsf{G}_1+\mathsf{G}_2+\mathsf{G}_3+\mathsf{G}_4+(\mathsf{G}_1\to\mathsf{G}_3)+(\mathsf{G}_2\to\mathsf{G}_3)+(\mathsf{G}_1\to\mathsf{G}_4)+(\mathsf{G}_2\to\mathsf{G}_4)\\ &\sqsupseteq\ \mathsf{G}_1+\mathsf{G}_2+\mathsf{G}_3+\mathsf{G}_4+(\mathsf{G}_1\to\mathsf{G}_3)+(\mathsf{G}_2\to\mathsf{G}_4)\ =\ (\mathsf{G}_1\circ\mathsf{G}_3)+(\mathsf{G}_2\circ\mathsf{G}_4).\end{aligned}$$

(Take also a look at FIGURE 1 on p. 9.)

Case 2: $\dot{\mathsf{G}}=\mathsf{G}\circ\ddot{\mathsf{G}}^\mathsf{T}$:

W.l.o.g. (see l. 2) assume now that $\ddot{\mathsf{G}}=\mathsf{G}_3+\mathsf{G}_4$ holds. Thence, we get $\dot{\mathsf{G}}=\mathsf{G}\circ(\mathsf{G}_3^\mathsf{T}+\mathsf{G}_4^\mathsf{T})$ where $\dot{\mathsf{G}}\sqsupseteq\mathsf{G}_1+\mathsf{G}_2$. This enables us to choose G' and G'' such that, w.l.o.g., $\mathsf{G}_1=\mathsf{G}'\circ\mathsf{G}_3^\mathsf{T}$ and $\mathsf{G}_2=\mathsf{G}''\circ\mathsf{G}_4^\mathsf{T}$ hold. (In other cases, G_3 and G_4 are only changed; however, note that by commutativity of $+$ this is not a problem: exchange G_1 with G_2!)

For $\mathsf{G}\circ(\mathsf{G}_3^\mathsf{T}+\mathsf{G}_4^\mathsf{T})\sqsupseteq\mathsf{G}_1+\mathsf{G}_2=(\mathsf{G}'\circ\mathsf{G}_3^\mathsf{T})+(\mathsf{G}''\circ\mathsf{G}_4^\mathsf{T})$ holds, we are done by l. 4.