Joint NCC & IIS Bull., Comp. Science, 13 (2000), 88–104

Algebraic specifications for dataflow computations design*

A. Votintseva, A. Yakovlev

1. Introduction

Dataflow networks are a well-known mathematical tool extensively used for representing, analyzing and modelling concurrent computing systems and their software. Formal dataflow models reported in the literature may be divided in two groups — static and dynamic. Static models [4] admit at most one token on an arc. This assumption severely limits the possibilities of concurrency. Dynamic models are free from this restriction due to program code copying [10] and token coloring [2, 11].

To get better understanding of the nature of concurrent computations, different approaches to representation of the semantics of dataflow networks presented in the literature have been studied. In the classical work by Kahn [6], the denotational semantics of dataflow computations was represented by fixed point equations. Using the Kahn principle, article [12] set forth the denotational semantics of real-time dataflow networks. The operational semantics in terms of firing sequences was given for static dataflow networks in [4] and for dynamic ones in [2, 11]. The possibility of modelling the operational semantics of static dataflow networks by ACP-terms was presented in [3]. Modularity and the Kahn principle were investigated in [8, 9] for dataflow networks whose semantics was represented by pomsets and trace languages. A fully abstract trace-model for dataflow networks was given in [5]. In paper [1] the event structure semantics was developed for a class of colored dataflow networks and formal relationships were established between a number of semantics notions, such as firing sequences, trace languages, dependence graphs and event structures. Our aim is to establish a full correspondence between these two models. We use algebraic specifications for this purpose. In [1] the class of well-formed colored dataflow networks was defined using algebraic operations. For the model of event structures we take the process algebra BPA (Basic Process Algebra) as a starting point. We consider an extension of BPA with a parallel operation $||$ and a binary Kleene star $*$, and denote it as $PBPA^*$. We adapt operators from $PBPA^*$ to prime event structures and obtain a class of well-formed event structures which is shown to be corresponding to well-formed colored dataflow networks.

The paper is organized as follows. In Section 2, we present the basic terminology concerning event structures. Section 3 defines the structure of a coloured dataflow network (for brevity, c-network). Then the operational semantics of a marked c-network in terms of firing sequences is given. In Section 4, we introduce a notion of decorated event structures, taking into account a context presented in computations, e.g., colors, time or something else. Further the notion of well-formed event structures is presented and the correspondence between the model just mentioned and well-formed c-networks is established. In the final section, some concluding remarks are given.

2. Basic notions of the event structures theory

Event structures have been firstly introduced in [7] being represented via sets of events with relations expressing causal dependences and conflicts between them. The subsets of events representing executions in the event structure are called configurations. They have to be conflict-free and left-closed with respect to the causality relation (all prerequisites for any event occurring in the execution must also occur).

*This work is supported by the RFBR (grant No 00-01-00898) and the youth grant 2000 of SB RAS

Definition 2.1. A *labeled prime event structure* over an alphabet Act (event structure for brevity) is a quadruple $S = (E, \leq, \#, l)$, where

(i) E is a countable set of events;

(ii) $\leq \subseteq E \times E$ is a partial order (the *causality relation*) satisfying the principle of finite causes: $\forall e \in E \ . \ \{d \in E \mid d \leq e\}$ is finite;

(iii) $\# \subseteq E \times E$ is a symmetric and irreflexive relation (the *conflict relation*) satisfying the principle of conflict heredity: $\forall e_1, e_2, e_3 \in E : (e_1 \leq e_2 \ \& \ e_1 \ \# \ e_3) \Rightarrow (e_2 \ \# \ e_3)$;

(iv) $l : E \to Act$ is a labeling function.

For an event structure $S = (E, \leq, \#, l)$, we assume that
$id = \{(e, e) \mid e \in E\}$;
$< \, = \, \leq \setminus id$;
$<^2 \subseteq <$ (transitivity);
$\lessdot \, = < \setminus <^2$ (immediate neighbourhood);
$e \ \#_1 \ d \iff e \ \# \ d \ \& \ \forall e_1, d_1 \in E \ . \ (e_1 \leq e \ \& \ d_1 \leq d), e_1 \# \ d_1 \Rightarrow (e_1 = e \ \& \ d_1 = d)$ (minimal conflict);
$\smile = \subseteq E \times E \setminus (< \cup <^{-1} \cup id \cup \#)$, $e^\bullet = \{e' \in E \mid e \lessdot e'\}$, ${}^\bullet e = \{e' \in E \mid e' \ \lessdot e\}$.
Two event structures are called isomorphic ($\mathcal{E} \cong \mathcal{F}$) iff there is a bijection between their sets of events $E_\mathcal{E}$ and $E_\mathcal{F}$ preserving the relations $\leq$ and $\#$ and labeling l.
The restriction of $\mathcal{E}$ to $C \subseteq E_\mathcal{E}$ is defined as $\mathcal{E} \lceil \mathcal{C} = (C, \leq_\mathcal{E} \cap (C \times C), \#_\mathcal{E} \cap (C \times C), l_\mathcal{E}|_C)$.

We consider an algebraic specification to select a class of event structures corresponding to a class of colored dataflow networks. As a representative we introduce a variant $PBPA^*$ of the known system BPA (Basic Process Algebra from [13]) with the parallel operator $||$ and the binary Kleene star $*$. We choose the process algebra BPA, since it has been proved in [14] that the axioms of BPA^* (i.e., BPA extended by the binary Kleene star) can completely characterize bisimilarity between processes. The $PBPA^*$ terms specify structures with all basic relations inherent in concurrent systems that can be represented by the model of event structures. Thus,

- the operator "$||$" means that all events in the first component are concurrent to all events in the second one;
- the operator "+" means that all events of one system are in conflict with all events of another one;
- the operator ";" means that all events of the first component are causally precedent to all events of the second one;
- the operator "$*$" means infinite iteration of two conflicting components with fixpoint semantics.

We define a set of conflict-free terms over an alphabet Act as follows:

$$PBPA^*_{cf}(Act) = a \mid (\alpha \ || \ \beta) \mid (\alpha; \beta), \text{ where } a \in Act, \alpha, \beta \in PBPA^*_{cf}(Act)$$

Then the following rules specify the set of $PBPA^*$-terms over the alphabet Act:

$$PBPA^*(Act) = a \mid (\alpha || \beta) \mid (\alpha + \beta) \mid (\gamma; \beta) \mid (\gamma * \beta),$$

where $a \in Act, \alpha, \beta \in PBPA^*(Act)$ and $\gamma \in PBPA_{cf}(Act)$.
Now we can define the event structure $\mathcal{E}_{PBPA^*}(p) = (E, \leq, \#, l)$ for a term $p \in PBPA^*(Act)$ by induction on the term construction:

1. Let $p = a \in Act$. Then $\mathcal{E}_{PBPA^*}(p) = (\{e\}, \emptyset, \emptyset, \{(e, a)\}$;
2. Let $p = p_1 \ || \ p_2$ with $\mathcal{E}_1 = \mathcal{E}_{PBPA^*}(p_1)$ and $\mathcal{E}_2 = \mathcal{E}_{PBPA^*}(p_2)$. Then

$$\mathcal{E}_{PBPA^*}(p) = (E_{\mathcal{E}_1} \cup E_{\mathcal{E}_2}, \leq_{\mathcal{E}_1} \cup \leq_{\mathcal{E}_2}, \#_{\mathcal{E}_1} \cup \#_{\mathcal{E}_2}, l_{\mathcal{E}_1} \cup l_{\mathcal{E}_2};$$

3. Let $p = p_1 + p_2$ with $\mathcal{E}_1 = \mathcal{E}_{PBPA^*}(p_1)$ and $\mathcal{E}_2 = \mathcal{E}_{PBPA^*}(p_2)$. Then

$$\mathcal{E}_{PBPA^*}(p) = (E_{\mathcal{E}_1} \cup E_{\mathcal{E}_2}, \leq_{\mathcal{E}_1} \cup \leq_{\mathcal{E}_2}, \#_{\mathcal{E}_1} \cup \#_{\mathcal{E}_2} \cup \{(e_1, e_2), (e_2, e_1) \mid e_1 \in E_{\mathcal{E}_1}, e_2 \in E_{\mathcal{E}_2}\}, l_{\mathcal{E}_1} \cup l_{\mathcal{E}_2};$$

4. Let $p = p_1;p_2$ with $\mathcal{E}_1 = \mathcal{E}_{PBPA^*}(p_1)$ and $\mathcal{E}_2 = \mathcal{E}_{PBPA^*}(p_2)$. Then

$$\mathcal{E}_{PBPA^*}(p) = (E_{\mathcal{E}_1} \cup E_{\mathcal{E}_2}, \leq_{\mathcal{E}_1} \cup \leq_{\mathcal{E}_2} \cup \{(e_1, e_2) \mid e_1 \in E_{\mathcal{E}_1}, e_2 \in E_{\mathcal{E}_2}\}, \#_{\mathcal{E}_1} \cup \#_{\mathcal{E}_2}, l_{\mathcal{E}_1} \cup l_{\mathcal{E}_2};$$

5. Let $p = p_1 * p_2$, $p^{(0)} = p_1 + p_2$ and $p^{(i+1)} = p_1;p^{(i)} + p_2$. Then $\mathcal{E}_{PBPA^*}(p)$ is defined as the minimal event structure such that $\mathcal{E}_{PBPA^*}(p^{(n)}) \sqsubseteq \mathcal{E}_{PBPA^*}(p)$ for all $n \in \mathbf{N}$.

By construction of $PBPA^*$-terms it is clear that $\mathcal{E}_{PBPA^*}(p)$ is a prime event structure for all $p \in PBPA^*(Act)$. Further we establish a correspondence between a class of $PBPA^*$-event structures (i.e., event structures constructed for $PBPA^*$-terms with the above rules) and the class of colored dataflow networks introduced in [1].

Definition 2.2. A *configuration* of an event structure $S = (E, \leq, \#, l)$ is a subset $X \subseteq E$ such that:
(i) $\forall e, e' \in X \ \diamond \ \neg(e \# e')$ (conflict-free);
(ii) $\forall e, e' \in E \ \diamond \ e \in X \ \& \ e' \leq e \Rightarrow e' \in X$ (left-closed).

We shall denote by $\mathcal{C}(S)$ the set of configurations of an event structure S.

In the grafical representation of an event structure, only immediate conflicts — not the inherited ones — are pictured. The immediate neighbourhood relation is represented by arcs, omitting those derivable by transitivity. Following these conventions, an example of an event structure is shown in Fig. 2.1.

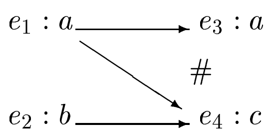


Figure 2.1.

3. Coloured dataflow networks

In this section, we consider coloured dataflow networks [1] and their operational semantics in terms of firing sequences.

A dataflow network (a network) is characterized by nodes and arcs. The nodes consist of links and actors. There are four kinds of actor nodes (operators, deciders, gates and colour actors) and two kinds of link nodes (data and control links). The arcs connecting links with actors and actors with links are called data and control arcs according to the type of link. The nodes and arcs are represented by two sets N and E which have to be nonempty, finite and disjoint.

Definition 3.1. A *network* is a pair $\mathcal{N} = (N, E)$, where

(i) N is a set of nodes consisting of a subset A of actors and a subset L of links with $A \cap L = \emptyset$. The actors are of the following types: operators (A^F), deciders (A^R), gates (A^G) and colour actors ($A^C = New \cup Next \cup Old$). The links are of two types: data links (L^I) and control links (L^R);

(ii) $E \subseteq ((A \cup \omega) \times L) \ \cup \ (L \times (A \cup \omega))$ is a set of arcs consisting of a subset E^I of data arcs and a subset E^R of control arcs. Here $\omega \notin N$ and $E^I \cap E^R = \emptyset$.

Fig. 3.1(a) shows the types of links. The types of actors are shown in Fig. 3.1(b):

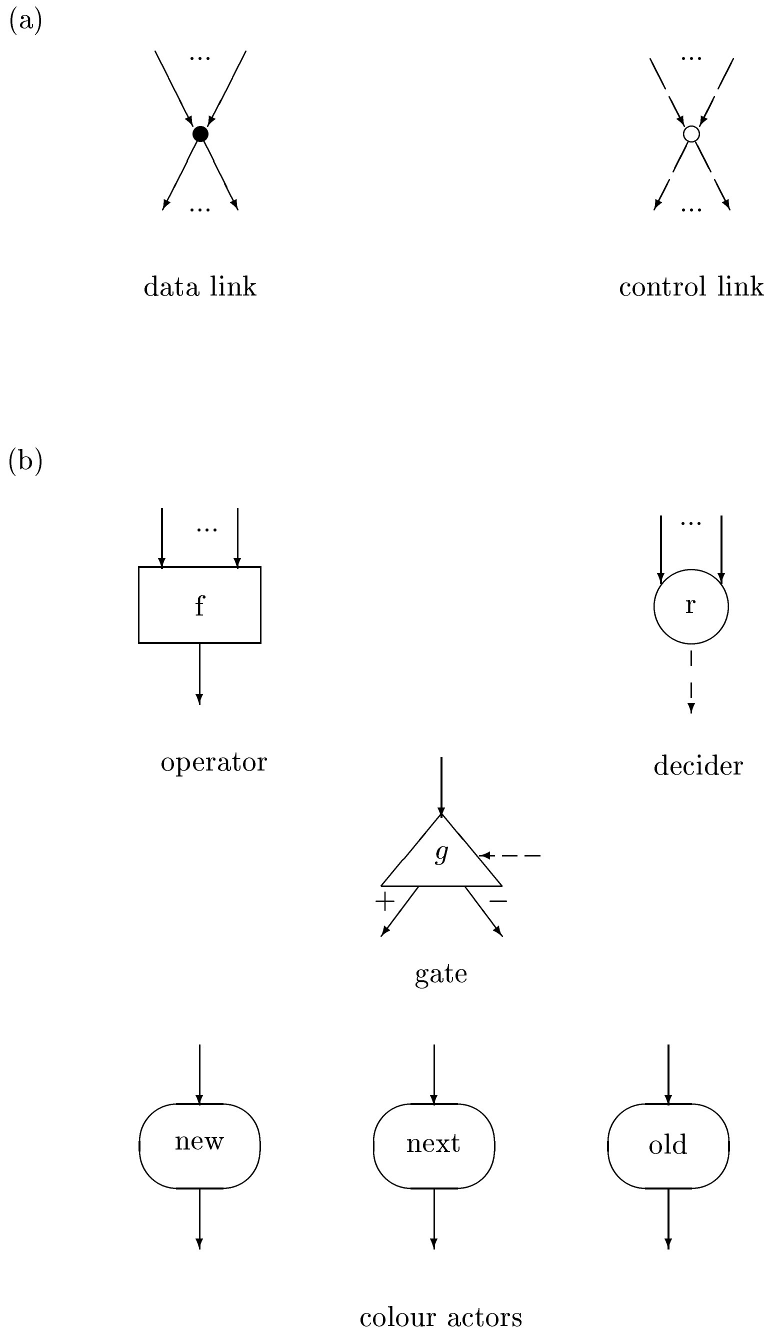


Figure 3.1.

At least one data (control) arc must terminate on and at least one data (control) arc must originate at each data (control) link. An operator has an ordered set of input data arcs and a single output data arc. A decider has an ordered set of input data arcs and a single output control arc. A gate has one input data arc and one input control arc and two output data arcs labelled by '+'- and '–'-signs. A colour actor: either $new \in New$ or $next \in Next$ or $old \in Old$ (allowing loop computations to be modelled). A colour actor has a single input data arc and a single output data arc.

For a node $n \in N$, we use $in(n)$ and $out(n)$ to denote the set of its input arcs and the set of its output arcs, respectively. We suppose $In = \{(\omega, l) \in E \mid l \in L\}$ (the set of input arcs of $\mathcal{N}$) and $Out = \{(l, \omega) \in E \mid l \in L\}$ (the set of output arcs of $\mathcal{N}$). We also fix $L^{In} = \{l \in L \mid (\omega, l) \in In\}$ (the set of input link nodes of $\mathcal{N}$) and $L^{Out} = \{l \in L \mid (l, \omega) \in Out\}$ (the set of output link nodes of $\mathcal{N}$).

For two nodes $n, n' \in A$, we shall write $n \hookrightarrow n'$, iff there exists $l \in L$ such that $out(n) \cap in(l) \neq \emptyset$ & $out(l) \cap in(n') \neq \emptyset$. By $n \stackrel{*}{\hookrightarrow} n'$ we denote the following fact: $n \hookrightarrow n_1$ & ... & $n_m \hookrightarrow n'$, where $n, n' \in (A^F \cup A^R)$, $\{n_1, \ldots, n_m\} = * \subseteq A \setminus (A^F \cup A^R)$ and $m \geq 0$.

The components of a network $\mathcal{N} = (N, E)$ are subscribed by the index $\mathcal{N}$, for example: $N_{\mathcal{N}}$ and $E_{\mathcal{N}}$. If clear from the context, the index $\mathcal{N}$ is omitted.

In order to get a class of networks suitable for our purpose, we introduce a notion of a *well-formed network*. Before doing so we need to define some additional notions and notations.

We consider the following elementary networks: operational (o-networks) $\hat{f}$, alternative (a-networks) $\stackrel{+}{r}$ and iterative (i-networks) $\stackrel{*}{r'}$ shown in Fig. 3.2. a), b) and c), respectively.

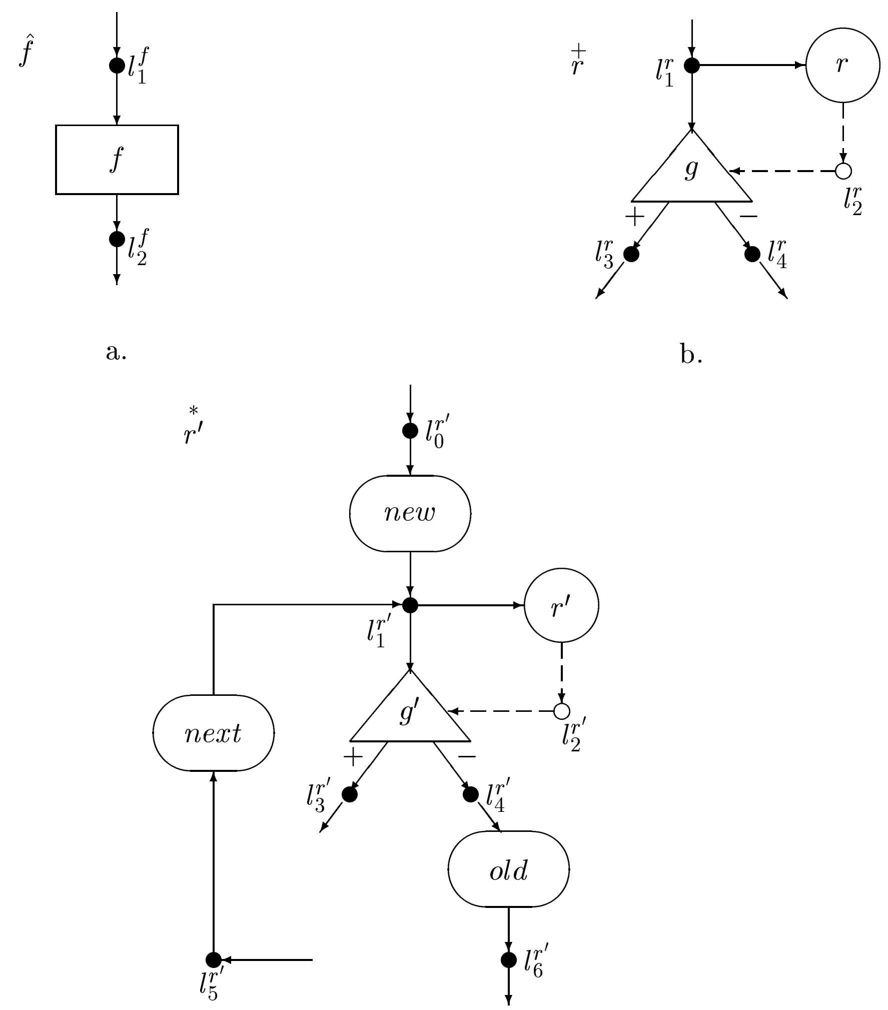


Figure 3.2.

Let $\mathcal{O}$ denote the set of all o-networks, $\mathcal{A}$ denote the set of all a-networks and $\mathcal{I}$ denote the set of all i-networks.

Now we can specify operations over nets:

(i) $||$-*operation* (parallel composition)
Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be networks such that $(N_{\mathcal{N}_1} \cap N_{\mathcal{N}_2}) = \emptyset$ and $(E_{\mathcal{N}_1} \cap E_{\mathcal{N}_2}) = \emptyset$. Then $(\mathcal{N}_1 \ || \ \mathcal{N}_2) = (N, E)$ is defined as: $N = N_{\mathcal{N}_1} \cup N_{\mathcal{N}_2}$; $E = E_{\mathcal{N}_1} \cup E_{\mathcal{N}_2}$.

(ii) *μ-operation* (merging of links)
Let $\mathcal{N}_1$ be a network and $\widehat{L} = \{l_1, \ldots, l_m\} \subseteq L^I_{\mathcal{N}_1}$. Let $l = <l_1, \ldots, l_m>$ be a link such that $in(l) = \cup(in(l_i) \mid l_i \in \widehat{L})$ and $out(l) = \cup(out(l_i) \mid l_i \in \widehat{L})$. Then $\mu(\mathcal{N}_1, \widehat{L}) = (N, E, T, C)$ is defined as:

$N = (A_{\mathcal{N}_1} \cup ((L_{\mathcal{N}_1} \setminus \widehat{L}) \cup \{l\}))$;

$$E = (E_{\mathcal{N}_1} \setminus (in(l) \cup out(l))) \cup \\ \{(n,l) \mid n \in N_{\mathcal{N}_1} \ \&\ \exists l_i \in \widehat{L} \ .\ (n, l_i) \in E_{\mathcal{N}_1})\} \cup \\ \{(l,n) \mid n \in N_{\mathcal{N}_1} \ \&\ \exists l_i \in \widehat{L} \ .\ (l_i, n) \in E_{\mathcal{N}_1})\} \cup \\ \{(\omega,l) \mid \forall l_i \in \widehat{L} \ .\ (\omega, l_i) \in E_{\mathcal{N}_1})\} \cup \\ \{(l,\omega) \mid \forall l_i \in \widehat{L} \ .\ (l_i, \omega) \in E_{\mathcal{N}_1})\}.$$

(iii) *;-operation* (sequential composition)
Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be networks such that $(N_{\mathcal{N}_1} \cap N_{\mathcal{N}_2}) = \emptyset$, $(E_{\mathcal{N}_1} \cap E_{\mathcal{N}_2}) = \emptyset$ and $\mid Out_{\mathcal{N}_1} \mid = \mid In_{\mathcal{N}_2} \mid = 1$.
Then $(\mathcal{N}_1;\mathcal{N}_2) = \mu((\mathcal{N}_1 \mid\mid \mathcal{N}_2), (L^{Out}_{\mathcal{N}_1} \cup L^{In}_{\mathcal{N}_2}))$.

(iv) *+-operation* (alternative composition)
Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be networks and $\overset{+}{r}$ be an a-network with a decider r such that $(N_{\mathcal{N}_1} \cap N_{\mathcal{N}_2}) \cup (N_{\mathcal{N}_1} \cap N_{\underset{r}{+}}) \cup (N_{\mathcal{N}_2} \cap N_{\underset{r}{+}}) = \emptyset$, $(E_{\mathcal{N}_1} \cap E_{\mathcal{N}_2}) \cup (E_{\mathcal{N}_1} \cap E_{\underset{r}{+}}) \cup (E_{\mathcal{N}_2} \cap E_{\underset{r}{+}}) = \emptyset$, $\mid In_{\mathcal{N}_1} \mid = \mid In_{\mathcal{N}_2} \mid = 1$ and $\mid Out_{\mathcal{N}_1} \mid = \mid Out_{\mathcal{N}_2} \mid = 1$. Let $g \in A^G_{\underset{r}{+}}$ and $l^r_3, l^r_4 \in L^I_{\underset{r}{+}}$ be such that (g, l^r_3) is the output '+'-arc and (g, l^r_4) is the output '–'-arc of g.
Then $(\mathcal{N}_1 + \mathcal{N}_2)^r = \mu(\mu(\mu(((\mathcal{N}_1 \mid\mid \mathcal{N}_2) \mid\mid \overset{+}{r}), (L^{In}_{\mathcal{N}_1} \cup \{l^r_3\})), (L^{In}_{\mathcal{N}_2} \cup \{l^r_4\})), (L^{Out}_{\mathcal{N}_1} \cup L^{Out}_{\mathcal{N}_2}))$.

(v) *$*$-operation* (iterative composition)
Let $\mathcal{N}_1$ be a network and $\overset{*}{r}{}'$ be an i-network with a decider r' such that $(N_{\mathcal{N}_1} \cap N_{\underset{r'}{*}}) = \emptyset$, $(E_{\mathcal{N}_1} \cap E_{\underset{r'}{*}}) = \emptyset$, $\mid In_{\mathcal{N}_1} \mid = \mid Out_{\mathcal{N}_1} \mid = 1$. Let $g \in A^G_{\underset{r'}{*}}$, $next \in A^C_{\underset{r'}{*}}$, $l^{r'}_3, l^{r'}_5 \in L^I_{\underset{r'}{*}}$ be such that $(g, l^{r'}_3)$ is the output '+'-arc of g and $(l^{r'}_5, next)$ is the output arc of $next$.
Then $(*\mathcal{N}_1)^{r'} = \mu(\mu((\mathcal{N}_1 \mid\mid \overset{*}{r}{}'), (L^{In}_{\mathcal{N}_1} \cup \{l^{r'}_3\})), (L^{Out}_{\mathcal{N}_1} \cup \{l^{r'}_5\}))$.

The examples of applying the operations (i)—(v) to nets are shown in Appendix.

We call a net $\mathcal{N}$ *well-formed* if it is defined as follows:

$$\mathcal{N} = \hat{f} \mid (\mathcal{N}_1 \mid\mid \mathcal{N}_2) \mid (\mathcal{N}_1;\mathcal{N}_2) \mid (\mathcal{N}_1 + \mathcal{N}_2)^r \mid (*\mathcal{N}_1))^{r'}, \text{ where}$$
$$\mathcal{N}_1, \mathcal{N}_2 \text{ are well-formed networks, } \hat{f} \in \mathcal{O},\ \overset{+}{r} \in \mathcal{A}\ \overset{*}{r}{}' \in \mathcal{I}.$$

We are now ready to define a notion of a coloured network.

Definition 3.2. A *coloured network* (c-network for brevity) is a quadruple $\mathcal{CN} = (\mathcal{N}, T, \Sigma, C)$, where

(i) $\mathcal{N}$ is a well-formed network;

(ii) T is a set of tokens consisting of two disjoint subsets T^I (data tokens) and T^R (control tokens) $(T^I \cap T^R = \emptyset)$;

(iii) Σ is a set of colours. Each colour is a triple $c = (x, y, z)$, where $c.x$ is the unique name of the loop, $c.y$ is the number of the loop iteration, and $c.z$ is the context that may be a colour itself;

(iv) $C : T \longrightarrow \Sigma$ is a colour function.

The components of a c-network $\mathcal{CN} = (\mathcal{N}, T, \Sigma, C)$ are subscribed by the index $\mathcal{CN}$. If clear from the context, the index $\mathcal{CN}$ is omitted. As an example, the well-formed network $\mathcal{CN}_0$ (with $\mathcal{N}_{\mathcal{CN}_0} = ((*\widehat{f}_1)^r; \widehat{f}_2))$ is shown in Fig. 3.3.

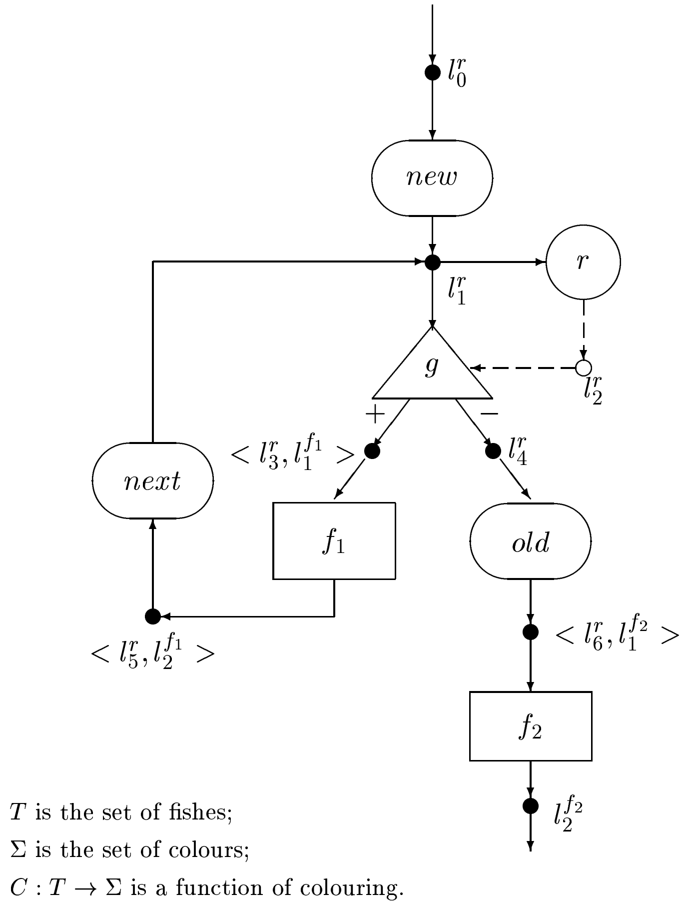


Figure 3.3.

For the sake of convenience, we fix a c-network $\mathcal{CN} = (\mathcal{N}, T, \Sigma, C)$ (with $\mathcal{N} = (N, E)$) and work with it throughout the paper.

A *marking* of $\mathcal{CN}$ is a function M defined from E into 2^T such that the tokens must have a type identical to the type of the arc. A marking M is called *initial* (denoted as M_{in}), iff
(i) $\forall e \in In\ \forall t, t' \in M(e) \ \diamond\ (t \neq t' \Rightarrow C(t) \neq C(t'))$;
(ii) $\forall e \in (E \setminus In) \ \diamond\ M(e) = \emptyset$.
Let a pair $(\mathcal{CN}, M_{in})$ denote the *initially marked c-network.*

We further specify an interpretation for $(\mathcal{CN}, M_{in})$ in order to provide a complete representation of the modelled computation.

Definition 3.3. An *interpretation* I of $(\mathcal{CN}, M_{in})$ is defined as follows:

(i) a domain $\mathbf{D}$ of values,

(ii) an assignment of a total function $\varphi : \mathbf{D}^m \longrightarrow \mathbf{D}$ to each operator $f \in A^F$, where $m =\mid in(f) \mid$,

(iii) an assignment of a total predicate $\psi : \mathbf{D}^m \longrightarrow \{\mathbf{true}, \mathbf{false}\}$ to each decider $r \in A^R$, where $m =\mid in(r) \mid$.

Besides we introduce a valuation function V that assigns a value $V(t) \in \mathbf{D}$ ($V(t) \in \{\mathbf{true}, \mathbf{false}\}$, respectively) to each token $t \in T^I$ ($t \in T^R$, respectively). Let $\mathbf{I}$ denote the set of all possible interpretations of $(\mathcal{CN}, M_{in})$. We use a triple $(\mathcal{CN}, M_{in}, I)$ to denote the interpreted $(\mathcal{CN}, M_{in})$.

For $(\mathcal{CN}, M_{in}, I)$, we define the *firing rule* associated with a node n as follows:

(i) An actor (a link) n is *enabled with a colour* c in a marking M if there is a token with a colour c on each (at least one) input arc of n.

(ii) An actor (a link) n enabled with a colour c in a marking M may be choosen to fire with a colour c yielding a new marking M' specified as follows:

1. One token with a colour c is removed from each (one) input arc of an actor (a link) n.
2. The tokens are added to the output arcs of n in the following way:

2.1. For a data (control) link n, one data (control) token with the colour $U(n,c)$ and the value $V(t)$ is added to each output arc of n, where t is the data (control) token removed from an input arc of n.

2.2. For an operator (a decider) n, one data (control) token with the colour $U(n,c)$ and the value $\varphi(V(t_1), \ldots, V(t_m))$ $(\psi\ (V(t_1), \ldots, V(t_m)))$ is added to the output arc of n, where $t_1, \ldots, t_m$ are the data tokens removed from the input arcs of n and $m = \mid in(n) \mid$.

2.3. For a gate n, one data token with the colour $U(n,c)$ and the value $V(t_1)$ is added to the '+'-arc, if $V(t_2) = \mathbf{true}$, or to the '–'-arc, if $V(t_2) = \mathbf{false}$, where t_1 and t_2 are respectively the data and control tokens removed from the input arcs of n.

2.4. For a colour actor n, one data token with the colour $U(n,c)$ and the value $V(t)$ is added to the output arc of n, where t is the data token removed from the input arc of n.

Here

$$U(n,c) = \begin{cases} (n,0,c), & \text{if } n \in New, \\ (c.x, c.y+1, c.z), & \text{if } n \in Next, \\ c.z, & \text{if } n \in Old, \\ c, & \text{otherwise.} \end{cases}$$

For $(\mathcal{CN}, M_{in}, I)$, a *firing* with a colour c of a node n is defined as a triple $M \xrightarrow{(n,c)} M'$ such that a transition from the marking M to the marking M' is consistent with the firing rule associated with n. We shall denote a firing with a colour c of a node n as just a pair (n,c), if information about markings M and M' is not significant.

From now on, we shall use $R = (N \times \Sigma)$ and $R' = ((A^F \cup A^R) \times \Sigma)$.

A *firing sequence* in $(\mathcal{CN}, M_{in}, I)$ is a string ρ over the alphabet R defined as:

(i) $\rho = \Lambda$ is a firing sequence in $(\mathcal{CN}, M_{in}, I)$ and $M_{in} \xrightarrow{\Lambda} M_{in}$,

(ii) Suppose ρ' is a firing sequence in $(\mathcal{CN}, M_{in}, I)$, $M_{in} \xrightarrow{\rho'} M$ and $M \xrightarrow{(n,c)} M'$, then $\rho = \rho'(n,c)$ is a firing sequence in $(\mathcal{CN}, M_{in}, I)$ and $M_{in} \xrightarrow{\rho'(n,c)} M'$.

Let $\mathbf{R}(\mathcal{CN}, M_{in}, I)$ denote the set of all firing sequences in $(\mathcal{CN}, M_{in}, I)$, and $\mathbf{R} = \cup(\mathbf{R}(\mathcal{CN}, M_{in}, I) \mid I \in \mathbf{I})$.

The specification of the behaviour of an initially marked c-net $(\mathcal{CN}, M_{in})$ by means of a labeled prime event structure has been described in [1]. To sketch it out, we first need to introduce the notion of a dependence graph and its projection defined for a firing sequence.

Definition 3.4.
Let $\rho \in \mathbf{R}$. Then the *D-graph* associated with ρ is the triple $\widetilde{G}_\rho = (\widetilde{V}_\rho, \widetilde{E}_\rho, \widetilde{l}_\rho)$ such that:

$\rho = \Lambda$**.** Then $\widetilde{G}_\Lambda = (\emptyset, \emptyset, \emptyset)$.

$\rho \neq \Lambda$. Let $\rho = \rho'(n,c)$ and assume that $\widetilde{G}_{\rho'} = (\widetilde{V}_{\rho'}, \widetilde{E}_{\rho'}, \tilde{l}_{\rho'})$ is defined. Then $\widetilde{G}_\rho = (\widetilde{V}_\rho, \widetilde{E}_\rho, \tilde{l}_\rho)$ for
$\widetilde{V}_\rho = \widetilde{V}_{\rho'} \cup \{((n,c),X)\}$, where
$X = \{((n',c'),X') \in \widetilde{V}_{\rho'} \mid out(n') \cap in(n) \neq \emptyset \text{ if } U(n',c') = c\}$,
$\widetilde{E}_\rho = \widetilde{E}_{\rho'} \cup (X \times \{((n,c),X)\})$ and
$\forall ((n',c'),X') \in \widetilde{V}_\rho \diamond \tilde{l}_\rho(((n',c'),X')) = (n',c')$.

In order to generalize some insignificant dependences in a d-graph, we define the notion of its *projection* onto R'.

Definition 3.5.
Let $\rho \in \mathbf{R}$. Then the *projection* of D-graph $\widetilde{G}_\rho = (\widetilde{V}_\rho, \widetilde{E}_\rho, \tilde{l}_\rho)$ on the set R' is the triple $\widehat{G}_\rho = (\widehat{V}_\rho, \widehat{E}_\rho, \hat{l}_\rho)$ such that:
$\widehat{V}_\rho = \{((n,c),X) \in \widetilde{V}_\rho \mid (n,c) \in R'\}$;
$\widehat{E}_\rho \subseteq \widehat{V}_\rho \times \widehat{V}_\rho$ such that $(((n,c),X),((n',c'),X')) \in \widehat{E}_\rho \iff$
$\exists (((n,c),X),((n_1,c_1),X_1)), \ \ldots \ (((n_m,c_m),X_m),((n',c'),X')) \in \widetilde{E}_\rho$
such that $(n,c),(n',c') \in R'$, $(n_1,c_1),\ldots,(n_m,c_m) \in (R \setminus R')$ if $m \geq 1$;
$\forall ((n',c'),X') \in \widehat{V}_\rho \diamond \ \hat{l}_\rho(((n',c'),X')) = (n',c')$.

Now we can give the event structure semantics for c-networks from [1].

Definition 3.6.
The event structure for $(\mathcal{CN}, M_{in})$ is a quadruple $\mathcal{E}((\mathcal{CN}, M_{in})) = (E, \leq, \#, l)$, where

- $E = \cup_{\rho \in \mathbf{R}} (\widehat{V}_\rho)$;
- $\leq = \cup_{\rho \in \mathbf{R}} (\widehat{E}_\rho)^*$;
- $\forall ((n,c),X),((n',c'),X') \in E \diamond \ ((n,c),X) \ \# \ ((n',c'),X') \ \Leftrightarrow \ \forall \rho \in \mathbf{R} \diamond$ $\{((n,c),X),((n',c'),X')\} \not\subseteq \widehat{V}_\rho$;
- $\forall ((n,c),X) \in E \diamond l_{\mathcal{E}(\mathcal{N})}(((n,c),X)) = (n,c)$.

Let us consider the initially marked c-net $(\mathcal{CN}_0, M_{in})$ (shown on Fig. 3.4), where $M_{in}((\omega, l_0^r)) = \{t\}$. Fig. 3.4 shows the final fragment of the event structure for $(\mathcal{CN}_0, M_{in})$, where $c_0 = (0,0,0)$, $c_1 = (new, 0, c_0)$, $c_2 = (new, 1, c_0)$ and $c_3 = (new, 2, c_0)$.

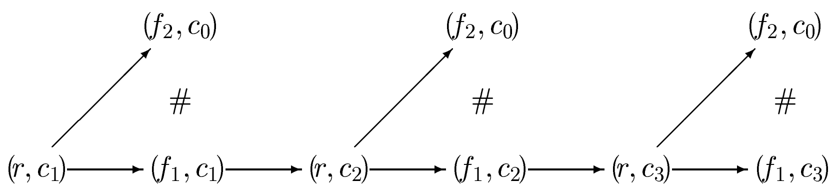


Figure 3.4.

4. Well-formed event structures

In this section we introduce a new version of event structures, namely, the context event structures over a separated alphabet $\widehat{Act} = Act^f \cup Act^r \cup \{\surd\}$, where $Act^f \cap Act^r = \emptyset$. Actions from Act^f are called basic and actions from Act^r are auxiliary for constructing well-formed event structures. We superinduce the notion of a context which is a countable set. When modeling a data domain by event structures, the context plays the role of an additional attribute, for example, time limitation or the set of colours.

Definition 4.1. *A context event structure* (over an alphabet $\widehat{Act} = Act^f \cup Act^r \cup \{\surd\}$ with a context $\mathcal{K}$) is a 5-tuple $\mathcal{E} = (E, <, \#, l, c)$, where

- E is a countable set of events;
- $< \subseteq E \times E$ is a (nonreflexive) partial order *causality relation*;
- $\# \subseteq E \times E$ is a symmetric nonreflexive relation (*conflict relation);*
- $l : E \to \widehat{Act}$ *is a labeling function;*
- $c : E \to \mathcal{K}$ *is a context assigning function.*

We use $E^t_{\mathcal{E}} = \{e \in E_{\mathcal{E}} \mid l_{\mathcal{E}}(e) = \surd\}$ to denote the set of terminate events, $E^{nt}_{\mathcal{E}} = E_{\mathcal{E}} \setminus E^t_{\mathcal{E}}$ to denote the set of nonterminate events, and $min\mathcal{E} = \{e \in E_{\mathcal{E}} \mid {}^{\bullet}e = \emptyset\}$ to denote the set of minimal elements.

We now introduce the notions of a free term p, the set of its actions $Var(p)$, and the class of isomorphic context event structures $[p]$ determined by this term. Let $Term^{\mathcal{K}}(Act)$ denote the set of free terms over the alphabet $Act = Act^F \cup Act^r$ with the context $\mathcal{K}$ constructed as follows.

1. $p = \mathbf{O} \in Term^{\mathcal{K}}(\widehat{Act})$, $Var(p) = \emptyset$ and $\mathcal{E} \in [p] \Rightarrow \mathcal{E} = (\{e\}, \emptyset, \emptyset, \{(e, \surd)\}, \{(e, k_0)\})$.
2. $p = a \in Act^f \Rightarrow p \in Term^{\mathcal{K}}(\widehat{Act})$, $Var(p) = \{a\}$ and

$$\mathcal{E} \in [p] \Rightarrow (\{e, e'\}, \{(e, e')\}, \emptyset, \{(e, a), (e', \surd)\}, \{(e, k_0), (e', k_0)\}).$$

Let $p, q \in Term^{\mathcal{K}}(\widehat{Act})$ such that $Var(p) \cap Var(q) = \emptyset$. Then the following operators are used to build terms :

(A) Parallel composition.
$p \mid\mid q \in Term^{\mathcal{K}}(\widehat{Act})$, $Var(p \mid\mid q) = Var(p) \cup Var(q)$ and $\mathcal{E} \in [p \mid\mid q] \Rightarrow \exists \mathcal{E}_1 \in [p], \mathcal{E}_2 \in [q]$: $E_{\mathcal{E}_1} \cap E_{\mathcal{E}_2} = \emptyset$ and $E_{\mathcal{E}} = E_{\mathcal{E}_1} \cup E_{\mathcal{E}_2}$, $<_{\mathcal{E}} = <_{\mathcal{E}_1} \cup <_{\mathcal{E}_2}$, $\#_{\mathcal{E}} = \#_{\mathcal{E}_1} \cup \#_{\mathcal{E}_2}$, $l_{\mathcal{E}} = l_{\mathcal{E}_1} \cup l_{\mathcal{E}_2}$, $c_{\mathcal{E}} = c_{\mathcal{E}_1} \cup c_{\mathcal{E}_2}$

(B) Sequential composition.
$p; q \in Term^{\mathcal{K}}(\widehat{Act})$, $Var(p; q) = Var(p) \cup Var(q)$ and $\mathcal{E} \in [p; q] \Rightarrow$
$\Rightarrow \exists \mathcal{E}_0 \in [p], \mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_n \in [q]$, where $n = \mid E^t_{\mathcal{E}_0} \mid$, such that $E^t_{\mathcal{E}_0} = \{e_1, e_2, \ldots, e_n\}$, $min\mathcal{E}_i = \{e_i\}$, for all $0 \leq i \leq n$, $E_{\mathcal{E}_i} \cap E_{\mathcal{E}_j} = \emptyset for 0 \leq i \neq j \leq n$, $E_{\mathcal{E}_0} \cap E_{\mathcal{E}_i} = \{e_i\}$ for $1 \leq i \leq n$ and $E_{\mathcal{E}} = \bigcup_{0 \leq i \leq n} E_{\mathcal{E}_i}$. Then:
$<_{\mathcal{E}} = \bigcup_{0 \leq i \leq n} <_{\mathcal{E}_i} \bigcup \{(e, e') \mid e \in E_{\mathcal{E}_0}, e' \in E_{\mathcal{E}_i} \forall 1 \leq i \leq n\}$,
$\#_{\mathcal{E}} = \bigcup_{0 \leq i \leq n} \#_{\mathcal{E}_i} \bigcup \{(e, e'), (e', e) \mid e \in E_{\mathcal{E}_0}, e' \in E_{\mathcal{E}_i}$ and $e \#_{\mathcal{E}_0} e_i, 1 \leq i \leq n\} \bigcup \{(e, e') \mid e \in E_{\mathcal{E}_i}, e' \in E_{\mathcal{E}_j}$ for $1 \leq i, j \leq n, i \neq j\}$, $l_{\mathcal{E}} = \bigcup_{1 \leq i \leq n} l_{\mathcal{E}_i} \bigcup l_{\mathcal{E}_0} \mid_{E^{nt}_{\mathcal{E}_0}}$, $c_{\mathcal{E}} = \bigcup_{1 \leq i \leq n} c_{\mathcal{E}_i} \bigcup c_{\mathcal{E}_0} \mid_{E^{nt}_{\mathcal{E}_0}}$.

(C) Alternative composition.
$p \overset{a}{+} q \in Term^{\mathcal{K}}(\widehat{Act})$ with $a \in Act^r \setminus (Var(p) \cup Var(q))$;
$Var(p \overset{a}{+} q) = Var(p) \cup Var(q) \cup \{a\}$ and $\mathcal{E} \in [p \overset{a}{+} q] \Rightarrow \exists \mathcal{E}_1 \in [p], \mathcal{E}_2 \in [q], d \notin E_{\mathcal{E}_1} \cup E_{\mathcal{E}_2}$:
$E_{\mathcal{E}_1} \cap E_{\mathcal{E}_2} = \emptyset$,
$\mid min\mathcal{E}_1 \mid = \mid min\mathcal{E}_2 \mid = 1$ and
$E_{\mathcal{E}} = E_{\mathcal{E}_1} \cup E_{\mathcal{E}_2} \cup \{d\}$,
$<_{\mathcal{E}} = <_{\mathcal{E}_1} \cup <_{\mathcal{E}_2} \cup \{(d, e) \mid e \in E_{\mathcal{E}_1} \cup E_{\mathcal{E}_2}\}$,
$\#_{\mathcal{E}} = \#_{\mathcal{E}_1} \cup \#_{\mathcal{E}_2} \cup \{(e, e') \mid e \in E_{\mathcal{E}_i}, e' \in E_{E_j}, i \neq j\}$,
$l_{\mathcal{E}} = l_{\mathcal{E}_1} \cup l_{\mathcal{E}_2} \cup \{(d, a)\}$,
$c_{\mathcal{E}} = c_{\mathcal{E}_1} \cup c_{\mathcal{E}_2} \cup \{(d, k_0)\}$.

(D) Context substitution.
$p = q[k'_1/k_1, \ldots, k'_n/k_n] \in Term^{\mathcal{K}}(\widehat{Act})$, where $k_i, k'_i \in \mathcal{K}$ for $1 \leq i \leq n$. Then $Var(p) = Var(q)$ and $\mathcal{E} \in [p] \Rightarrow \exists \mathcal{E}' \in [q] : \mathcal{E} = (E_{\mathcal{E}'}, <_{\mathcal{E}'}, \#_{\mathcal{E}'}, l_{\mathcal{E}'}, c)$,
with

$$c(e) = \begin{cases} k'_i, & \text{if } c_{\mathcal{E}'}(e) = k_i \text{ for some } 1 \leq i \leq n \text{ ;} \\ c_{\mathcal{E}'}(e), & \text{otherwise.} \end{cases}$$

for all $e \in E_{\mathcal{E}'}$.

(E) Iterative composition.
$p \overset{a}{*} \in Term^{\mathcal{K}}(\widehat{Act})$, where $a \in Act^r \backslash Var(p)$; $Var(p \overset{r}{*}) = Var(p) \cup \{a\}$ and $\mathcal{E} \in [p \overset{r}{*}] \Rightarrow \mathcal{E}$ is a minimal context event structure such that $\forall n \in \mathbf{N} \exists \mathcal{E}_n \in [p^{(n)}]$: $\mathcal{E}_n \sqsubseteq \mathcal{E}$, ($\mathcal{E}_n$ is a substructure of $\mathcal{E}$), where the term $p^{(n)}$ is built inductively as follows :
$p^{(0)} = \mathbf{O}$,
$p^{(n)} = (\mathbf{O} \overset{a}{+} (p; p^{(n-1)}))[k_a^1/k_0, k_a^2/k_a^1, \ldots, k_a^n/k_a^{n-1}]$.

In the set of free terms we consider a subset of basic terms $Term_0^{\mathcal{K}}(\widehat{Act}) \subset Term^{\mathcal{K}}(\widehat{Act})$:

$$Term_0^{\mathcal{K}}(\widehat{Act}) \;=\; b \;|\; (p \,||\, q) \;|\; (p; q) \;|\; (p \overset{a}{+} q) \;|\; (p \overset{a}{*}), \text{ where} \quad b \in Act^f,\ a \in Act^r,\ p, q \in Term_0^{\mathcal{K}}(\widehat{Act}).$$

Definition 4.2. A context event structure $\mathcal{E}$ is called a *well-formed event structure* if $\mathcal{E} \in [p]$ for some $p \in Term_0^{\mathcal{K}}(\widehat{Act})$.

Two context event structures $\mathcal{E}_1$ and $\mathcal{E}_2$ with contexts $\mathcal{K}_1$ and $\mathcal{K}_2$, respectively, are called isomorphic ($\mathcal{E}_1 \cong \mathcal{E}_2$) iff there is an isomorphism f between $\mathcal{E}_1$ and $\mathcal{E}_2$ with a substitution $h : \mathcal{K}_1 \to \mathcal{K}_2$, i.e. $f : E_{\mathcal{E}_1} \to E_{\mathcal{E}_2}$ is a bijection such that $\forall e, d \in E_{\mathcal{E}_1} : (e <_{\mathcal{E}_1} d \Leftrightarrow f(e) <_{\mathcal{E}_2} f(d))$ & $(e \#_{\mathcal{E}_1} d \Leftrightarrow f(e) \#_{\mathcal{E}_2} f(d))$ and $\forall e \in E_{\mathcal{E}_1}, d \in E_{\mathcal{E}_2}$: $d = f(e) \Rightarrow (l_{\mathcal{E}_1}(e) = l_{\mathcal{E}_2}(d))$ & $c_{\mathcal{E}_1}(e) = h(c_{\mathcal{E}_2}(d)))$ It can be noted that the isomorphism just introduced for context event structures correlates with that considered in Section 2 for prime event structures, assuming that prime event structures have the null context $\mathcal{K}_{null} = \{\mathbf{O}\}$ (i.e., $k_a^i = k_0 = \mathbf{O}$ for all $i \in \mathbf{N}$ and $a \in Act^r$ s.t. $k_a^i \in \mathcal{K}_{null}$). Thus, we can consider an isomorphism between a context event structure $\mathcal{E}$ and a prime event structure $\mathcal{E}'$ ($\mathcal{E} \cong \mathcal{E}'$) with the substitution $h : \mathcal{K}_{\mathcal{E}} \to \mathcal{K}_{null}$. In other words, we can consider such isomorphisms simply without a substitution.

Proposition 4.1. Let $\mathcal{E}_0$ be a well-formed event structure (over the alphabet $\widehat{Act}$). Then there is $q \in PBPA^*(\widehat{Act})$ such that $\mathcal{E}_{PBPA^*}(q) \cong \mathcal{E}_0$.

Proof. We prove it by induction on the structure of a term p. Assume $p \in Term_0^{\mathcal{K}}(\widehat{Act})$ to be such that $\mathcal{E}_0 \in [p]$. We construct the corresponding $PBPA^*$-term q (considering an isomorphism without substitution by induction on the number of ;-operator used in the term p).

1. Suppose that the term p does not include a sequential composition. We prove the case by induction on the structure of p.
Four cases are possible:

$p = a \in Act^f$. Then clearly $q = (a; \surd) \in PBPA^*(\hat{Act})$. It is easy to see that $\mathcal{E}_{PBPA^*}(q) \cong \mathcal{E}_0$.

$p = (p_1 || p_2)$. Assume $q_1, q_2 \in PBPA^*(\widehat{Act})$ to be constructed so that $\mathcal{E}_{PBPA^*}(q_1) = \mathcal{F}_1 \cong \mathcal{E}_1 \in [p_1]$ and $\mathcal{F}_2 = \mathcal{E}_{PBPA^*}(q_2) \cong \mathcal{E}_2 \in [p_2]$. Then obviously $q = (q_1 || q_2) \in PBPA^*(\widehat{Act})$. It is necessary to show $\mathcal{E}_{PBPA^*}(q) \cong \mathcal{E}_0$. Let f_1 and f_2 be an isomorphism between $\mathcal{F}_2$ and $\mathcal{E}_2$. We can take $\mathcal{F}_1, \mathcal{F}_2, \mathcal{E}_1$ and $\mathcal{E}_2$ such that $E_{\mathcal{F}_1} \cap E_{\mathcal{F}_2} = \emptyset$ and $E_{\mathcal{E}_1} \cap E_{\mathcal{E}_2} = \emptyset$. It is clear that $f_1 \cup f_2 : E_{\mathcal{F}_1} \cup E_{\mathcal{F}_2} \to E_{\mathcal{E}_1} \cup E_{\mathcal{E}_2}$ is an isomorphism between $\mathcal{F} = \mathcal{E}_{PBPA^*}(q)$ and $\mathcal{E}_0$.

$p = p_1; p_2$. Assume $q_1, q_2 \in PBPA^*(\widehat{Act})$ to be constructed so that $\mathcal{E}_{PBPA^*}(q_1) = \mathcal{F}_1 \cong \mathcal{E}_1 \in [p_1]$ and $\mathcal{F}_2 = \mathcal{E}_{PBPA^*}(q_2) \cong \mathcal{E}_2 \in [p_2]$ and $a \in Act^r \subset \widehat{Act}$. Then $q = a; (q_1 + q_2) \in PBPA^*(\widehat{Act})$. It is necessary to show that $\mathcal{E}_{PBPA^*}(q) \cong \mathcal{E}_0$. Let f_1 be an isomorphism between $\mathcal{F}_1$ and $\mathcal{E}_1$ and f_2 be an isomorphism between $\mathcal{F}_2$ and $\mathcal{E}_2$. We consider $\mathcal{F}_1, \mathcal{F}_2, \mathcal{E}_1$ and $\mathcal{E}_2$ such that $E_{\mathcal{F}_1} \cap E_{\mathcal{F}_2} = E_{\mathcal{E}_1} \cap E_{\mathcal{E}_2} = \emptyset$ and $d \notin E_{\mathcal{E}_1} \cup E_{\mathcal{E}_2}$, where $d \in E_{\mathcal{E}_0}$ such that $l_{\mathcal{E}_0}(d) = a$. Let $e \notin E_{\mathcal{F}_1} \cup E_{\mathcal{F}_2}$ be such that $l_{\mathcal{E}_{PBPA^*}(q)}(e) = a$. Then it is clear that $f = f_1 \cup f_2 \cup \{(e, d)\}$ is an isomorphism between $\mathcal{E}_{PBPA^*}(q)$ and $\mathcal{E}_0$.

$p = (p_1 \overset{a}{*})$. Assume $q_1 \in PBPA^*(\widehat{Act})$ to be constructed so that $\mathcal{E}_{PBPA^*}(q_1) = \mathcal{F}_1 \cong \mathcal{E}_1 \in [p_1]$. Then $q = a; (q_1; a * \surd) \in PBPA^*(\widehat{Act})$. It is necessary to show that $\mathcal{E}_{PBPA^*}(q) \cong \mathcal{E}_0$. Let us consider the sequence of $PBPA^*$-terms:

$q^{(0)} = \surd$,
$q^{(1)} = a; (q_1; \surd + \surd)$,
$q^{(2)} = a; (q_1; (a; (q_1; \surd + \surd)) + \surd), \ldots, q^{(n)} = a; (q_1; q^{(n-1)} + \surd)$. It is easy to see that $q = lim_{n\to\infty} q^{(n)}$ and $\mathcal{E}_{PBPA^*}(q^{(i)}) \cong \mathcal{E}_n \in [p^{(n)}]$. Hence, $\mathcal{E}_{PBPA^*}(q)$ is the minimal event structure such that $\mathcal{E}_{PBPA^*}(q) \cong \mathcal{E}_0$..

2. Let $p = (p_1; p_2)$. Since the term p_2 includes less sequential composition operators than p, we can construct the term $q_2 \in PBPA^*(\widehat{Act})$ such that $\mathcal{E}_{PBPA^*}(q_2) \cong \mathcal{E}_2 \in [p_2]$. We build the term $q \in PBPA^*(\widehat{Act})$ corresponding to $p \in Term_0^{\mathcal{K}}(\widehat{Act})$ by induction on the structure of p_1. It is clear from the definition of the sequential composition that $(p_1; p_2); p_3 = p_1; (p_2; p_3) \forall p_1, p_2, p_3 \in Term_0^{\mathcal{K}}(\widehat{Act})$. Thus, the following four cases are only worth to be considered.

$p_1 = a \in Act^f$. Then $q = (a; q_2) \in PBPA^*(\widehat{Act})$

$p_1 = p'_1 \,||\, p''_1$. This case is invalid since $\forall \mathcal{E}_1 \in [p_1]$: $| min\mathcal{E}_1 | \geq 2$ due to the definition of the parallel composition. This contradicts the sequential composition for $Term_0^{\mathcal{K}}(\widehat{Act})$.

$p_1 = (p'_1 \overset{a}{+} p''_1)$. Since p'_1 and p''_1 are less than p_1, we can construct the terms $q'_1, q''_1 \in PBPA^*(\widehat{Act})$ such that $\mathcal{E}_{PBPA^*}(q'_1) \cong \mathcal{E}_1 \in [p'_1; p_2]$ and $\mathcal{E}_{PBPA^*}(q''_1) \cong \mathcal{E}_1 \in [p''_1; p_2]$. Let us consider the term $q = a; (q'_1 + q''_1) \in PBPA^*(\widehat{Act})$. By the reasoning analogous to that in case 1(c) of the present proof, one can establish that $\mathcal{E}_{PBPA^*}(q) \cong \mathcal{E}_0$.

$p_1 = (p_0 \overset{a}{*})$. Since the term $p_0 \in Term_0^{\mathcal{K}}(\widehat{Act})$ is less than p_1, we can construct the term $q_0 \in PBPA^*(\widehat{Act})$ such that $\mathcal{E}_{PBPA^*}(q_0) \cong \mathcal{E}_1 \in [(p_0; a)]$. Let us consider the term $q = a; (q_0 * q_2)$. By the reasoning analogous to that in case 1(d) (replacing q_2 instead of $\surd$) of the present proof, one can establish that $\mathcal{E}_{PBPA^*}(q) \cong \mathcal{E}_0$. □

From Proposition 4.1 it obviously follows that a well-formed event structure is a prime event structure, since $\mathcal{E}_{PBPA^*}(p)$ is a prime event structure for all $p \in PBPA^*(\widehat{Act})$.

The action $\surd$ is not significant indeed, since it is only used to denote a possible exit from an iterative cycle in a well-formed event structure. By this reason we consider the notion of weak isomorphism defined as follows: $\mathcal{E} \cong_\omega \mathcal{F}$ (this means that $\mathcal{E}$ and $\mathcal{F}$ are weakly isomorphic) if $\mathcal{E}|_{E^{nt}_{\mathcal{E}}} \cong \mathcal{F}|_{E^{nt}_{\mathcal{F}}}$, what means that this equivalence notion takes into consideration only non-terminate events.

Theorem 4.1.
(i) Let $(\mathcal{CN}, M_{in})$ be an initially marked c-network such that $\forall e \in In : t \in M_{in}(e) \Rightarrow C(t) = c_0$. Then there is a well-formed event structure $\mathcal{E}_0$ such that $\mathcal{E}_0 \cong_\omega \mathcal{E}(\mathcal{CN}, M_{in})$.
(ii) Let $\mathcal{E}_0$ be a well-formed event structure over the alphabet $\widehat{Act}$ with a context $\mathcal{K}$. Then there is an initially marked c-network $(\mathcal{CN}, M_{in})$, where $\forall e \in In : t \in M_{in}(e) \Rightarrow C(t) = c_0$ such that $\mathcal{E}(\mathcal{CN}, M_{in}) \cong_\omega \mathcal{E}_0$.

Proof.
(i) We take the following alphabet $Act^r = A^R$, $Act^f = A^F$ and a substitution h such that $k_0 = h(c_0)$ and $k_a^i = h(new(a), i, c)$, where $new(a) \in New$ such that $new(a) \hookrightarrow a$. We prove the case by induction on the structure of the formula for $\mathcal{N}_{\mathcal{CN}}$:

1. Let $\mathcal{N} = \hat{f}$. Then for any $\mathcal{E} \in [f]$ it is obvious that $\mathcal{E}_0 \cong_\omega \mathcal{E}(\mathcal{CN}, M_{in})$.

2. Let $\mathcal{N} = \mathcal{N}_{\mathcal{CN}_1} \,||\, \mathcal{N}_{\mathcal{CN}_2}$. By the induction hypothesis, there are terms p (for $\mathcal{N}_{\mathcal{CN}_1}$) and q (for $\mathcal{N}_{\mathcal{CN}_2}$) and well-formed event structures $\mathcal{E}_1 \in [p]$ and $\mathcal{E}_2 \in [q]$ such that $E_{\mathcal{E}_1} \cap E_{\mathcal{E}_2} = \emptyset$ and $\mathcal{E}_1 \cong_\omega \mathcal{E}(\mathcal{CN}_1, M_{in})$, $\mathcal{E}_2 \cong_\omega \mathcal{E}(\mathcal{CN}_2, M_{in})$. Then, clearly, $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \in [p \,||\, q] \in Term^{\mathcal{K}}(\widehat{Act})$ built according to the rule **(A)** satisfies the following: $\mathcal{E} \cong_\omega \mathcal{E}(\mathcal{CN}_1, M_{in}) \cup \mathcal{E}(\mathcal{CN}_2, M_{in}) = \mathcal{E}(\mathcal{CN}, M_{in})$.

3. Let $\mathcal{N} = \mathcal{N}_{\mathcal{CN}_1}; \mathcal{N}_{\mathcal{CN}_2}$. By the induction hypothesis, there are terms p (for $\mathcal{N}_{\mathcal{CN}_1}$) and q (for $\mathcal{N}_{\mathcal{CN}_2}$) and well-formed event structures $\mathcal{E}_0 \in [p]$ and $\mathcal{E}_1, \ldots, \mathcal{E}_n \in [q]$ such that $E_{\mathcal{E}_i} \cap E_{\mathcal{E}_j} = \emptyset$, while $i \neq j$ and $\mathcal{E}_0 \cong_\omega \mathcal{E}(\mathcal{CN}_1, M_{in})$, $\mathcal{E}_i \cong_\omega \mathcal{E}(\mathcal{CN}_2, M_{in})$ for $1 \leq i \leq n$. Then, clearly, $\mathcal{E} \in [p;q] \in Term^{\mathcal{K}}(\widehat{Act})$ built according to the rule **(B)** satisfies the following: $\mathcal{E} \cong_\omega \mathcal{E}(\mathcal{CN}, M_{in})$.

4. Let $\mathcal{N} = (\mathcal{N}_{\mathcal{CN}_1} + \mathcal{N}_{\mathcal{CN}_2})^r$. By the induction hypothesis, there are terms p (for $\mathcal{N}_{\mathcal{CN}_1}$) and q (for $\mathcal{N}_{\mathcal{CN}_2}$) and well-formed event structures $\mathcal{E}_1 \in [p]$ and $\mathcal{E}_2 \in [q]$ such that $E_{\mathcal{E}_1} \cap E_{\mathcal{E}_2} = \emptyset$ and $\mathcal{E}_1 \cong_\omega \mathcal{E}(\mathcal{CN}_1, M_{in})$, $\mathcal{E}_2 \cong_\omega \mathcal{E}(\mathcal{CN}_2, M_{in})$. Then, clearly, $\mathcal{E} \in [p \overset{r}{+} q] \in Term^{\mathcal{K}}(\widehat{Act})$ built according to the rule **(C)** satisfies the following $\mathcal{E} \cong_\omega \mathcal{E}(\mathcal{CN}, M_{in})$.

5. $\mathcal{N} = (*\mathcal{N}_{\mathcal{CN}_1})^r$. By the induction hypothesis, there is a term p (for $\mathcal{N}_{\mathcal{CN}_1}$) and a well-formed event structure $\mathcal{E}_1 \in [p]$ such that $\mathcal{E}_1 \cong_\omega \mathcal{E}(\mathcal{CN}_1, M_{in})$. Then, clearly, $\mathcal{E} \in [p \overset{r}{*}] \in Term^{\mathcal{K}}(\widehat{Act})$ built according to the rule **(D)** satisfies the following: $\mathcal{E} \cong_\omega \mathcal{E}(\mathcal{CN}, M_{in})$.

(ii) We take $A^F = Act^f$, $A^R = Act^r$ and a substitution h built as follows: $h(c_0) = k_0$, $h(new_i, j, c) = k_a^j$ with $new_i \hookrightarrow a$. We prove the case by induction on the structure of p:

1. Let $p = a$. Then $\mathcal{N}_{\mathcal{CN}} = \hat{a}$. From the definition of o-nets we have

$$\mathbf{R}(\mathcal{CN}, M_{in}) = \{(l_1^a, c_0)(a, c_0)(l_2^a, c_0)\}.$$

Then, by construction of $\mathcal{E}(\mathcal{CN}, M_{in})$, we get the following $E_{\mathcal{E}(\mathcal{CN},M_{in})} = \{((a, c_0), \emptyset)\}$, $<_{\mathcal{E}(\mathcal{CN},M_{in})} = \emptyset$, $\#_{\mathcal{E}(\mathcal{CN},M_{in})} = \emptyset$, $l_{\mathcal{E}(\mathcal{CN},M_{in})}((a, c_0), \emptyset) = (a, c_0)$. Obviously, $\mathcal{E}(\mathcal{CN}, M_{in}) \cong_\omega \mathcal{E}_0$.

2. Let $p = p_1 \mid\mid p_2$. Then we have $\mathcal{N}_{\mathcal{CN}} = \mathcal{N}_{\mathcal{CN}_1} \mid\mid \mathcal{N}_{\mathcal{CN}_2}$, where $\mathcal{N}_{\mathcal{CN}_1}$ is a c-network for p_1 and $\mathcal{N}_{\mathcal{CN}_2}$ is a c-network fpr p_2. By the definition of the $||$-operation for networks, it follows that

$$\mathbf{R}(\mathcal{CN}, M_{in}) = \mathbf{R}(\mathcal{CN}_1, M_{in}) \cup \mathbf{R}(\mathcal{CN}_2, M_{in}).$$

Therefore,

$$\begin{aligned} E_{\mathcal{E}(\mathcal{CN},M_{in})} &= E_{\mathcal{E}(\mathcal{CN}_1,M_{in})} \cup E_{\mathcal{E}(\mathcal{CN}_2,M_{in})}, \\ <_{\mathcal{E}(\mathcal{CN},M_{in})} &= <_{\mathcal{E}(\mathcal{CN}_1,M_{in})} \cup <_{\mathcal{E}(\mathcal{CN}_2,M_{in})}, \\ \#_{\mathcal{E}(\mathcal{CN},M_{in})} &= \#_{\mathcal{E}(\mathcal{CN}_1,M_{in})} \cup \#_{\mathcal{E}(\mathcal{CN}_2,M_{in})}, \\ l_{\mathcal{E}(\mathcal{CN},M_{in})} &= l_{\mathcal{E}(\mathcal{CN}_1,M_{in})} \cup l_{\mathcal{E}(\mathcal{CN}_2,M_{in})}. \end{aligned}$$

By construction, it is easy to see that $\mathcal{E}(\mathcal{CN}, M_{in}) \cong_\omega \mathcal{E}_0$.

3. Let $p = p_1; p_2$. Then we have $\mathcal{N}_{\mathcal{CN}} = \mathcal{N}_{\mathcal{CN}_1}; \mathcal{N}_{\mathcal{CN}_2}$, where $\mathcal{N}_{\mathcal{CN}_1}$ is a c-network for p_1 and $\mathcal{N}_{\mathcal{CN}_2}$ is a c-network fpr p_2. By the definition of the ;-operation for networks, it follows that

$$\mathbf{R}(\mathcal{CN}, M_{in}) = max\mathbf{R}(\mathcal{CN}_1, M_{in}) \circ \mathbf{R}(\mathcal{CN}_2, M_{in}) = \{\rho\rho' \mid \rho \in max\mathbf{R}(\mathcal{CN}_1, M_{in}), \rho' \in \mathbf{R}(\mathcal{CN}_2, M_{in})\},$$

where

$$max\mathbf{R}(\mathcal{CN}_1, M_{in}) = \{\rho \in \mathbf{R}(\mathcal{CN}_1, M_{in}) \mid \forall \rho' \in \mathbf{R}(\mathcal{CN}_1, M_{in}) \diamond \rho' = \rho\rho'' \Rightarrow \rho'' = \Lambda\}$$

and $\mid max\mathbf{R}(\mathcal{CN}_1, M_{in}) \mid = n = \mid E^t_{\mathcal{E}_0} \mid$ for $\mathcal{E}_0 \in [p_1]$. Therefore,

$$E_{\mathcal{E}(\mathcal{CN},M_{in})} = E_{\mathcal{E}(\mathcal{CN}_1,M_{in})} \cup \bigcup_{1 \le i \le n} E_{\mathcal{E}_i(\mathcal{CN}_2,M_{in})},$$

where $E_{\mathcal{E}_i(\mathcal{CN}_2,M_{in})} \cap E_{\mathcal{E}_j(\mathcal{CN}_2,M_{in})} = \emptyset$ with $i \neq j$ and $\mathcal{E}_i(\mathcal{CN}_2, M_{in}) \cong \mathcal{E}_j(\mathcal{CN}_2, M_{in})$ for all $1 \le i, j \le n$;

$$\begin{aligned} <_{\mathcal{E}(\mathcal{CN},M_{in})} &= <_{\mathcal{E}(\mathcal{CN}_1,M_{in})} \cup \bigcup_{1 \le i \le n} <_{\mathcal{E}_i(\mathcal{CN}_2,M_{in})} \cup \{(e, e') \mid e \in E_{\mathcal{E}(\mathcal{CN}_1,M_{in})}, e' \in E_{\mathcal{E}_i(\mathcal{CN}_2,M_{in})} \text{ and} \\ &\quad e <_{\mathcal{E}(\mathcal{CN}_1,M_{in})} d_i \in maxE_{\mathcal{E}(\mathcal{CN}_1,M_{in})}\}; \\ \#_{\mathcal{E}(\mathcal{CN},M_{in})} &= \#_{\mathcal{E}(\mathcal{CN}_1,M_{in})} \cup \bigcup_{1 \le i \le n} \#_{\mathcal{E}_i(\mathcal{CN}_2,M_{in})} \cup \{(e, e') \mid e \in E_{\mathcal{E}_i(\mathcal{CN}_2,M_{in})}, e' \in E_{\mathcal{E}_j(\mathcal{CN}_2,M_{in})}, \\ &\quad i \neq j\} \cup \{(e, e'), (e', e) \mid e \in E_{\mathcal{E}(\mathcal{CN}_1,M_{in})}, e' \in E_{\mathcal{E}_i(\mathcal{CN}_2,M_{in})} \text{ and} \\ &\quad e \#_{\mathcal{E}(\mathcal{CN}_1,M_{in})} d_i \in maxE_{\mathcal{E}(\mathcal{CN}_1,M_{in})}\}; \\ l_{\mathcal{E}(\mathcal{CN},M_{in})} &= l_{\mathcal{E}(\mathcal{CN}_1,M_{in})} \cup \bigcup_{1 \le i \le n} l_{\mathcal{E}_i(\mathcal{CN}_2,M_{in})}. \end{aligned}$$

By construction, it is easy to see that $\mathcal{E}(\mathcal{CN}, M_{in}) \cong_\omega \mathcal{E}_0$.

4. Let $p = p_1 \overset{r}{+} p_2$. Then we have $\mathcal{N}_{\mathcal{CN}} = (\mathcal{N}_{\mathcal{CN}_1} + \mathcal{N}_{\mathcal{CN}_2})^r$, where $\mathcal{N}_{\mathcal{CN}_1}$ is a c-network for p_1 and $\mathcal{N}_{\mathcal{CN}_2}$ is a c-network fpr p_2. By the definition of the $+$-operation for networks, it follows that

$$\mathbf{R}(\mathcal{CN}, M_{in}) = \{(l_1^r, c)(r, c)(l_2^r, c)(g, c)(l, c)\rho \mid \rho \in \mathbf{R}(\mathcal{CN}_1, M_{in}) \cup \mathbf{R}(\mathcal{CN}_2, M_{in})$$

and

$$l \in \{< l_3^r, l_1^{\mathcal{N}_{\mathcal{CN}_1}} >, < l_4^r, l_1^{\mathcal{N}_{\mathcal{CN}_2}} >\}\}.$$

Therefore,

$$\begin{aligned}
E_{\mathcal{E}(\mathcal{CN}, M_{in})} &= E_{\mathcal{E}(\mathcal{CN}_1, M_{in})} \cup E_{\mathcal{E}(\mathcal{CN}_2, M_{in})} \cup \{((r, c), X)\}, \\
<_{\mathcal{E}(\mathcal{CN}, M_{in})} &= <_{\mathcal{E}(\mathcal{CN}_1, M_{in})} \cup <_{\mathcal{E}(\mathcal{CN}_2, M_{in})} \cup \{(e, e') \mid e' \in E_{\mathcal{E}(\mathcal{CN}_1, M_{in})} \cup E_{\mathcal{E}(\mathcal{CN}_2, M_{in})}, e = ((r, c), X)\}, \\
\#_{\mathcal{E}(\mathcal{CN}, M_{in})} &= \#_{\mathcal{E}(\mathcal{CN}_1, M_{in})} \cup \#_{\mathcal{E}(\mathcal{CN}_2, M_{in})} \cup \{(e, e'), (e', e) \mid e \in E_{\mathcal{E}(\mathcal{CN}_1, M_{in})}, e' \in E_{\mathcal{E}(\mathcal{CN}_2, M_{in})}\}, \\
l_{\mathcal{E}(\mathcal{CN}, M_{in})} &= l_{\mathcal{E}(\mathcal{CN}_1, M_{in})} \cup l_{\mathcal{E}(\mathcal{CN}_2, M_{in})} \cup \{(((r, c), X), (r, c))\}.
\end{aligned}$$

By construction, it is easy to see that $\mathcal{E}(\mathcal{CN}, M_{in}) \cong_\omega \mathcal{E}_0$.

5. Let $p = p_1 \overset{r}{*}$. Then we have $\mathcal{N}_{\mathcal{CN}} = (*\mathcal{N}_{\mathcal{CN}_1})^r$, where $\mathcal{N}_{\mathcal{CN}_1}$ is a c-network for p_1. By the definition of the $*$-operation for networks, it follows that

$$\mathbf{R}(\mathcal{CN}, M_{in}) = \bigcup_{i \in \mathbf{N}} \{\rho^i \mid \rho \in \mathbf{R}(\mathcal{CN}_1, M_{in})\},$$

where $\rho^0 = \Lambda$, $\rho^n = \rho\rho^{n-1}$. Therefore,

$$E_{\mathcal{E}(\mathcal{CN}, M_{in})} = \bigcup_{i \in \mathbf{N}} E_{\mathcal{E}_i(\mathcal{CN}_1, M_{in})},$$

where

$$E_{\mathcal{E}_i(\mathcal{CN}_1, M_{in})} \cap E_{\mathcal{E}_j(\mathcal{CN}_1, M_{in})} = \emptyset, \quad i \neq j$$

and

$$\begin{aligned}
\mathcal{E}_i(\mathcal{CN}_1, M_{in}) &\cong \mathcal{E}_j(\mathcal{CN}_1, M_{in}) \ \forall i \in \mathbf{N}, \\
<_{\mathcal{E}(\mathcal{CN}, M_{in})} &= \bigcup_{i \in \mathbf{N}} <_{\mathcal{E}_i(\mathcal{CN}_1, M_{in})} \cup \bigcup_{i<j} \{(e, e') \mid e \in E_{\mathcal{E}_i(\mathcal{CN}_1, M_{in})}, e' \in E_{\mathcal{E}_j(\mathcal{CN}_1, M_{in})}\}, \\
\#_{\mathcal{E}(\mathcal{CN}, M_{in})} &= \bigcup_{i \in \mathbf{N}} \#_{\mathcal{E}_i(\mathcal{CN}_1, M_{in})}, \\
l_{\mathcal{E}(\mathcal{CN}, M_{in})} &= \bigcup_{i \in \mathbf{N}} l_{\mathcal{E}_i(\mathcal{CN}_1, M_{in})}.
\end{aligned}$$

By construction, it is easy to see that $\mathcal{E}(\mathcal{CN}, M_{in}) \cong_\omega \mathcal{E}_0$. □

5. Conclusion

In this paper we have formalized a possibility for algebraic specification to establish a correspondence between event structures and c-networks. We propose a new variation of the event structure model. We have enriched this well-known formal model by adding a notion of context. This allows us to increase the expressiveness of the event structure model. Moreover, we define a number of algebraic operations over recently introduced context event structures which are shown to be corresponding to the operations of the earlier known algebra BPA^* defined over event structures. The main result of the paper establishes a mutual correspondence between the classes of these two models (coloured dataflow networks and context event structures) defined by algebraic operations.

It is worth remarking that the obtained results have been formulated in terms of finite objects (finite algebraic formulas representing infinite systems). Such investigations allow us to classify and unify different abstract models of concurrent processes. Further research could include different equivalence notions over c-networks and their relations to the similar ones over event structures.

References

[1] I. Virbitskaite, A. Votintseva, D. Chkliaev, *Investigating Semantic Notions for Coloured Dataflow Networks*, Hildesheimer Informatik-Bericht, **27/96**, Institut für Informatik, Universität Hildesheim, Hildesheim, Germany, August 1996, 31 p.

[2] Arvind, K.P. Gostelow, *U-interpretator*, Computer, **15**, No 2, 1982, 42–49.

[3] J.A. Bergstra, J.W. Klop *A process algebra for the operational semantics of static data flow networks*, Tech. Report IW 222/83, CWI, Amsterdam, 1983.

[4] J.B. Dennis, *Data flow schemata*, Lect. Notes Comput. Sci., **5**, 1972, 187–216.

[5] B.A. Jonson, *A fully abstract trace model for dataflow networks*, POPL'89, 1989, 155–165.

[6] G. Kahn, D.B. McQueen, *Coroutines and Networks of Parallel Processes*, Information Processing 77, Proc. of IFIP Congress 77, Toronto, August 8–12, 1977, North-Holland, 1977, 993–998.

[7] M. Nielsen, G. Plotkin, G. Winskel, *Petri nets, event structures and domains*, Theoret. Comput. Sci., **13**, No 1, 1981, 85–108.

[8] A. Rabinovich, *Pomset semantics is consistent with data flow semantics*, Bull. of EATCS, **32**, 1987.

[9] A. Rabinovich, B.A. Trakhtenbrot, *Nets of processes and data flow*, Lect. Notes Comput. Sci., **354**, 1988, 574–602.

[10] J. Rumbaugh, *A parallel asynchronous computer architecture for data flow programs*, Tech. Report TR-150, MIT, Project MAC, May 1975.

[11] A.H. Veen, *A formal model for data flow programs with token coloring*, Tech. Report IW 179, CWI, Amsterdam, 1981.

[12] R.K. Yetes, G.R. Gao, *A Kahn principle of networks of nonmonotonic real-time processes*, Lect. Notes Comput. Sci., **694**, 1993, 209–227.

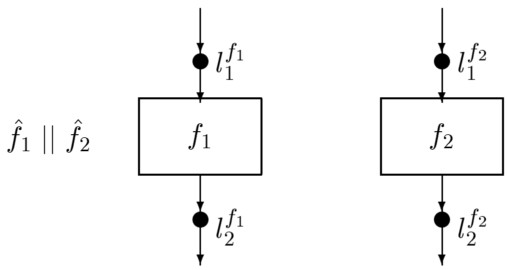
[13] J.C.M. Baeten, W.P. Weijland, *Process algebra*, Cambridge Tracts in Theor. Comput. Sci., Cambridge University Press, **18**, 1990.

[14] W.J. Fokkink, H. Zantema, *Basic process algebra with iteration: completeness of its equational axioms*, Comput. J., **37**, No 4, 1994, 259–267.

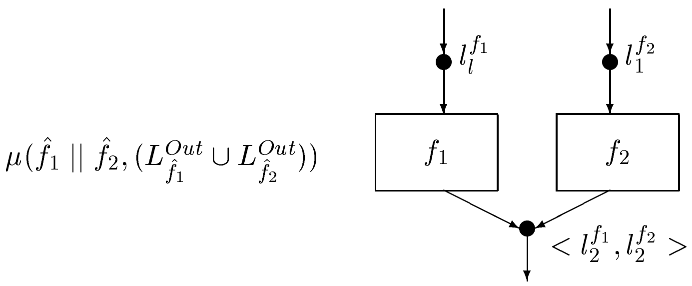
6. Appendix

The examples of the algebraic operations over dataflow networks.

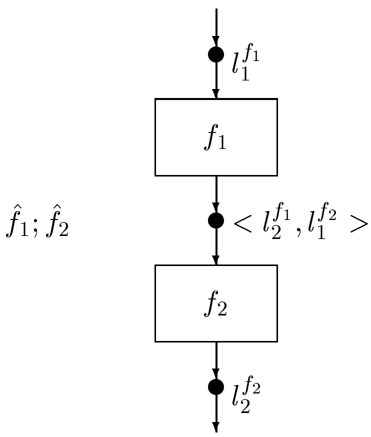
(a) $||$-*operation* (parallel composition)
An example of using the $||$-operation to o-net $\hat{f}_1$ and $\hat{f}_2$:



(b) μ-*operation* (merging of links)
An example of using the μ-operation to net $(\hat{f}_1 \,||\, \hat{f}_2)$ and set $(L_{f_1}^{Out} \cup L_{f_2}^{Out})$:



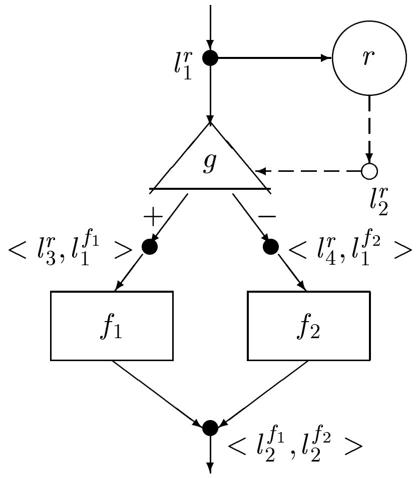
(c) ;-*operation* (sequential composition)
An example of using the ;-operation to o-net $\hat{f}_1$ and $\hat{f}_2$:



(d) +-*operation* (alternative composition)

An example of using the +-operation to o-net $\hat{f}_1$, $\hat{f}_2$ and a-net $\overset{+}{r}$:

$(\hat{f}_1 + \hat{f}_2)^r$



(e) $*$-*operation* (iterative composition)

An example of using the $*$-operation to o-net $\hat{f}$ and i-net $\overset{*}{r'}$:

$(*\hat{f})^{r'}$