

Computing with Natural Numbers in Cause-Effect Structures

Ludwik Czaja

¹Vistula University, Warsaw

²Institute of Informatics, The University of Warsaw

lczaja@mimuw.edu.pl

Abstract. Implementation of the standard model of natural numbers arithmetic is constructed in terms of cause-effect (c-e) structures. The numbers are represented by sets of (unstructured) tokens at nodes of the c-e structures, so the unary coding of numbers is applied. Such modelling of arithmetic operations is possible due to the inhibiting capability in the c-e structures. The exact cost of the typical operations modelled by c-e structures constructed in this paper, is computed.

1. A summary of elementary cause-effect structures

A complete presentation of elementary and extended cause-effect (c-e) structures is in [Cza 2019]. Here, some notions needed for demonstration of their capability of modelling computation with numbers are quoted.

Definition 1.1 (set $F[\mathbb{X}]$, quasi-semiring of formal polynomials)

Let \mathbb{X} be a non-empty enumerable set. Their elements, called *nodes*, are counterparts of places in Petri nets. $\theta \notin \mathbb{X}$ is a symbol called *neutral*. It plays role of neutral element for operations on terms, called *formal polynomials over \mathbb{X}* . The names of nodes, symbol θ , operators $+$, \bullet , called addition and multiplication, and parentheses are symbols out of which formal polynomials are formed. A node's name and θ is a formal polynomial; if K and L are formal polynomials, then $(K + L)$ and $(K \bullet L)$ are too. Their set is denoted by $F[\mathbb{X}]$. Assume stronger binding of \bullet than $+$, which allows for dropping some parentheses. Addition and multiplication of formal polynomials is defined as follows: $K \oplus L = (K + L)$, $K \odot L = (K \bullet L)$. To simplify notation, let us use $+$ and \bullet instead of \oplus and \odot . It is required that the system $\langle F[\mathbb{X}], +, \bullet, \theta \rangle$ obeys the following equality axioms for all $K, L, M \in F[\mathbb{X}]$, $x \in \mathbb{X}$:

- | | | | |
|----------------|---|------------------------------------|---|
| (+) | $\theta + K = K + \theta = K$ | (\bullet) | $\theta \bullet K = K \bullet \theta = K$ |
| (++) | $K + K = K$ | ($\bullet\bullet$) | $x \bullet x = x$ |
| (+++) | $K + L = L + K$ | ($\bullet\bullet\bullet$) | $K \bullet L = L \bullet K$ |
| (++++) | $K + (L + M) = (K + L) + M$ | ($\bullet\bullet\bullet\bullet$) | $K \bullet (L \bullet M) = (K \bullet L) \bullet M$ |
| (+ \bullet) | If $L \neq \theta \Leftrightarrow M \neq \theta$ then $K \bullet (L + M) = K \bullet L + K \bullet M$ | | |

Algebraic system which obeys these axioms will be referred to as a *quasi-semiring of formal polynomials*. □

Copyright © 2019 for this paper by its author. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Definition 1.2 (cause-effect structure, carrier, set \mathbf{CE})

A cause-effect structure (c-e structure) over \mathbb{X} is a pair $U = \langle C, E \rangle$ of total functions:

$$\begin{aligned} C: \mathbb{X} &\rightarrow \mathbf{F}[\mathbb{X}] && (\text{cause function; nodes occurring in } C(x) \text{ are causes of } x) \\ E: \mathbb{X} &\rightarrow \mathbf{F}[\mathbb{X}] && (\text{effect function; nodes occurring in } E(x) \text{ are effects of } x) \end{aligned}$$

such that x occurs in the formal polynomial $C(y)$ iff y occurs in $E(x)$. Carrier of U is the set $\text{car}(U) = \{x \in \mathbb{X} : C(x) \neq \theta \vee E(x) \neq \theta\}$. U is of finite carrier iff $|\text{car}(U)| < \infty$ ($|\dots|$ denotes cardinality). The set of all c-e structures over \mathbb{X} is denoted by $\mathbf{CE}[\mathbb{X}]$. Since \mathbb{X} is fixed, we write just \mathbf{CE} . \square

C and E are total, thus each c-e structure comprises all nodes from \mathbb{X} , also the isolated ones - those from outside of its carrier. In presenting c-e structures graphically, only their carriers are drawn. A representation of a c-e structure $U = \langle C, E \rangle$ as a set of annotated nodes is $\{x_{E(x)}^{C(x)} : x \in \text{car}(U)\}$. U is also presented as a directed graph with $\text{car}(U)$ as set of vertices labelled with objects of the form $x_{E(x)}^{C(x)}$ ($x \in \text{car}(U)$); there is an edge from x to y iff y occurs in the polynomial $E(x)$. Fig.1.1 (a) and (b) depict two graphical presentations of the same c-e structure with carrier $\{a, b, c, d, e, f, g, h\}$; in (a) the encircled nodes comprise groups making products in formal polynomials in (b), where the sums of the products create families of the groups. This c-e structure is the set $\{a_e^\theta, b_e^\theta, c_e^\theta, d_e^\theta, e_{f \cdot g + h}^{a+b \cdot c + c \cdot d}, f_\theta^e, g_\theta^e, h_\theta^e\}$.

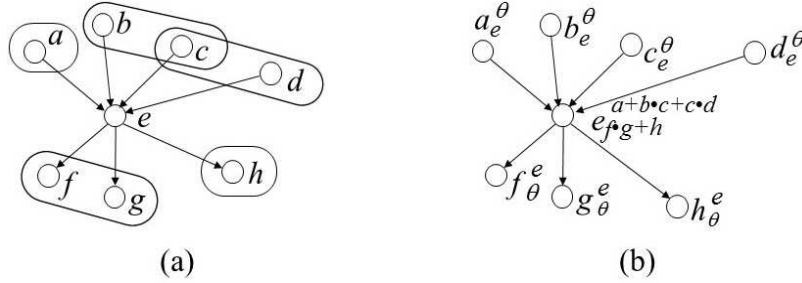


Fig.1.1 (a) Predecessors and successors of the node e , grouped into families: $\{\{a\}, \{b, c\}, \{c, d\}\}$ and $\{\{f, g\}, \{h\}\}$. (b) Notation by means of polynomials; the arrows are redundant: used only for graphic presentation of c-e structures

Definition 1.3 (addition and multiplication, monomial c-e structure)

For c-e structures $U = \langle C_U, E_U \rangle$, $V = \langle C_V, E_V \rangle$ define:
 $U + V = \langle C_{U+V}, E_{U+V} \rangle = \langle C_U + C_V, E_U + E_V \rangle$ where
 $(C_U + C_V)(x) = C_U(x) + C_V(x)$ and similarly for E
 $U \bullet V = \langle C_{U \bullet V}, E_{U \bullet V} \rangle = \langle C_U \bullet C_V, E_U \bullet E_V \rangle$ where
 $(C_U \bullet C_V)(x) = C_U(x) \bullet C_V(x)$ and similarly for E

(The same symbols "+" and "•" are used for operations on c-e structures and formal polynomials). U is a *monomial* c-e structure iff each polynomial $C_U(x)$ and $E_U(x)$ is a monomial, i.e. does not comprise non-reducible (relative to equations in Definition 1.1) operation "+". \square

Apart from the representation of c-e structures as a set $\{x_{E(x)}^{C(x)} : x \in \text{car}(U)\}$, their linear notation is used as the so-called "*arrow-expressions*": $\{x_y^\theta, y_\theta^x\}$ is an arrow, denoted as $x \rightarrow y$ and, consequently, $\{x_y^\theta, y_\theta^x\} \bullet \{y_z^\theta, z_\theta^y\} \bullet \{z_u^\theta, u_\theta^z\} \dots = \{x_y^\theta, y_z^x, z_u^y, u_\theta^z, \dots\}$ is a chain, denoted as $x \rightarrow y \rightarrow z \rightarrow u \dots$. Bidirectional arrow $x \leftrightarrow y$ denotes $x \rightarrow y \rightarrow x$ (equivalent to $y \rightarrow x \rightarrow y$), that is, the close cycle $\{x_y^y, y_x^x\}$. Chains and arrows in particular, may be combined into "arrow expressions" representing some c-e structures. For instance c-e structure $\{a_{x+y}^\theta, b_{x \bullet y}^\theta, x_\theta^{a \bullet b}, y_\theta^{a \bullet b}\}$ may be written as $(a \rightarrow x + a \rightarrow y) \bullet (b \rightarrow x) \bullet (b \rightarrow y)$.

The set \mathbf{CE} with addition, multiplication and a distinguished element denoted also by θ and understood as the empty c-e structure (θ, θ) , where θ is a constant function $\theta(x) = \theta$ for all $x \in \mathbb{X}$, makes an algebraic system similar to that in Definition 1.1, the quasi-semiring of c-e structures. Therefore, the system $\langle \mathbf{CE}[\mathbb{X}], +, \bullet, \theta \rangle$ obeys the following equations for all $U, V, W \in \mathbf{CE}[\mathbb{X}]$, $x, y \in \mathbb{X}$:

$$\begin{array}{ll}
(+)& \theta + U = U + \theta = U & (\bullet)& \theta \bullet U = U \bullet \theta = U \\
(++)& U + U = U & (\bullet\bullet)& (x \rightarrow y) \bullet (x \rightarrow y) = x \rightarrow y \\
(+++)& U + V = U + V & (\bullet\bullet\bullet)& U \bullet V = V \bullet U \\
(++++)& U + (V + W) = (U + V) + W & (\bullet\bullet\bullet\bullet)& U \bullet (V \bullet W) = (U \bullet V) \bullet W \\
(+\bullet)& \text{If } C_V(x) \neq \theta \Leftrightarrow C_W(x) \neq \theta \text{ and } E_V(x) \neq \theta \Leftrightarrow E_W(x) \neq \theta \text{ then} \\
& U \bullet (V + W) = U \bullet V + U \bullet W
\end{array}$$

Definition 1.4 (partial order \leq ; substructure, set $\mathbf{SUB}[V]$, firing component, set \mathbf{FC} , pre-set and post-set)

For $U, V \in \mathbf{CE}$ let $U \leq V \Leftrightarrow V = U + V$; \leq is a partial order in \mathbf{CE} . If $U \leq V$ then U is a *substructure* of V ; $\mathbf{SUB}[V] = \{U : U \leq V\}$ is the set of all substructures of V . For $A \subseteq \mathbf{CE}$: $V \in A$ is *minimal* (w.r.t. \leq) in A iff $\forall W \in A: (W \leq V \Rightarrow W = V)$.

A minimal in $\mathbf{CE} \setminus \{\theta\}$ c-e structure $Q = (C_Q, E_Q)$ is a *firing component* iff Q is a monomial c-e structure and $C_Q(x) = \theta \Leftrightarrow E_Q(x) \neq \theta$ for any $x \in \text{car}(Q)$. The set of all firing components is denoted by \mathbf{FC} , thus the set of all firing components of $U \in \mathbf{CE}$ is $\mathbf{FC}[U] = \mathbf{SUB}[U] \cap \mathbf{FC}$. Let for $Q \in \mathbf{FC}$ and $G \subseteq \mathbf{FC}$:

$$\begin{array}{ll}
\bullet Q & = \{x \in \text{car}(Q) : C_Q(x) = \theta\} & (\text{pre-set of } Q) \\
Q \bullet & = \{x \in \text{car}(Q) : E_Q(x) = \theta\} & (\text{post-set of } Q) \\
\bullet G & = \bigcup_{Q \in G} \bullet Q & (\text{pre-set of } G)
\end{array}$$

$$G^\bullet = \bigcup_{Q \in G} Q^\bullet \quad (\text{post-set of } G) \quad \square$$

Definition 1.5 (state of elementary c-e structures)

A *state* is a subset of the set of nodes: $s \subseteq \mathbb{X}$. The set of all states: $\mathbb{S} = 2^{\mathbb{X}}$. A node x is *active* at the state s iff $x \in s$ and *passive* otherwise. As in case of Petri nets, we say "x holds a token" when x is active. Obviously, the state might be defined equivalently as a two-valued function $s: \mathbb{X} \rightarrow \{0, 1\}$. \square

Definition 1.6 (sequential and parallel semantics of elementary c-e structures)

Sequential. For $Q \in \mathbf{FC}[U]$, $s \in \mathbb{S}$, let $[[Q]] \subseteq \mathbb{S} \times \mathbb{S}$ be a binary relation defined as:

$(s, t) \in [[Q]]$ iff $\bullet Q \subseteq s \wedge Q^\bullet \cap s = \emptyset \wedge t = (s \setminus \bullet Q) \cup Q^\bullet$ (Q transforms state s into t). *Semantics* $[[U]]$ of $U \in \mathbf{CE}$ is: $[[U]] = \bigcup_{Q \in \mathbf{FC}[U]} [[Q]]$. $[[U]]^*$ is

its reflexive and transitive closure, that is, $(s, t) \in [[U]]^*$ if and only if $s = t$ or there exists a sequence of states s_0, s_1, \dots, s_n with $s = s_0$, $t = s_n$ and $(s_j, s_{j+1}) \in [[U]]$ for $j = 0, 1, \dots, n-1$. State t is *reachable* from s in semantics $[[U]]$ and the sequence s_0, s_1, \dots, s_n is called a *computation* in U .

Parallel. Firing components Q and P are *detached* if and only if $\bullet Q \cap \bullet P = \emptyset$. For $G \subseteq \mathbf{FC}[U]$, $G \neq \emptyset$, containing only pairwise detached (*pwd*) firing components, let $[[G]]_{\text{par}} \subseteq \mathbb{S} \times \mathbb{S}$ be a binary relation defined as: $(s, t) \in [[G]]_{\text{par}}$ iff $\bullet G \subseteq s \wedge G^\bullet \cap s = \emptyset \wedge t = (s \setminus \bullet G) \cup G^\bullet$ (G transforms state s into t). *Semantics* $[[U]]_{\text{par}}$ of $U \in \mathbf{CE}$ is: $[[U]]_{\text{par}} = \bigcup_{G \subseteq \mathbf{FC}[U] \wedge G \text{ is } \text{pwd}} [[G]]_{\text{par}}$. Closure $[[U]]_{\text{par}}^*$ and reachability and computation are defined as in the sequential case. \square

Note that $[[U]] = \emptyset$ iff $\mathbf{FC}[U] = \emptyset$. Behaviour of elementary c-e structures may be thought of as a token game: if each node in a firing component's pre-set holds a token and none in its post-set does, then remove tokens from the pre-set and put them in the post-set. This is illustrated in Fig.1.2. A number of properties inferred from above definitions are in [Cza 2019].

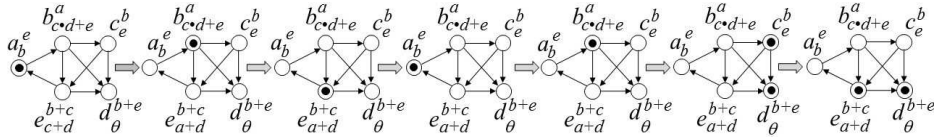


Fig.1.2. Example of activity (successive transformations) of a c-e structure.

2. A summary of extended cause-effect structures

The structural definition of extended c-e structures, their firing component in particular, is the same as in Definitions 1.1 - 1.4. The extensions consist in redefining the state, treating the pre and post sets of firing components as multisets, and redefining semantics. It is assumed that with a given c-e structure $U \in \mathbf{CE}[\mathbb{X}]$ (i.e. already constructed by operations introduced in Definition 1.3) and the set of its firing components $\mathbf{FC}[U] = \mathbf{SUB}[U] \cap \mathbf{FC}$, some additional informations are associated. The following extensions of elementary c-e structures endowed with such informations will be acquired: c-e structures with multi-valued nodes thus their capacities, coefficients of monomials in polynomials annotating nodes (counterparts of weight of arrows in place/transition Petri nets) and with inhibitors. To this end, a notation for multisets is convenient: let \mathbb{N} be the set of natural numbers including 0 and $\mathbb{N}_\omega = \mathbb{N} \cup \{\omega\}$, where the value ω means infinity, that is $\omega > n$ for each $n \in \mathbb{N}$. A *multiset* over a set X is a (total) function $f: X \rightarrow \mathbb{N}_\omega$. If the set $\{x: f(x) \neq 0\}$ is finite then the

linear-form notation is adopted for multisets, e.g.

X	a	b	c	d	e
$f(X)$	2	0	3	1	ω

is denoted as $2 \otimes a + 3 \otimes c + d + \omega \otimes e$. A multiset is *zero* \mathbb{O} , when $\mathbb{O}(x) = 0$ for all x . Addition, subtraction and multiplication on multisets is defined: $(f + g)(x) = f(x) + g(x)$, $(f - g)(x) = f(x) - g(x)$ for $f(x) \geq g(x)$, $(f \cdot g)(x) = f(x) \cdot g(x)$, comparison of multisets: $f \leq g$ iff $f(x) \leq g(x)$ for all x . Assume the customary arithmetic of ω : $\omega + n = \omega$, $\omega - n = \omega$, $\omega + \omega = \omega$ and additionally $0 - \omega = 0$.

Definition 2.1 (state)

A state of extended c-e structure U is a total function $\bar{s}: \mathit{car}(U) \rightarrow \mathbb{N}$, thus a multiset over $\mathit{car}(U)$. The set of all states of U is denoted by \mathbb{S} .

□

Definition 2.2. (weights of monomials and capacity of nodes)

For a c-e structure $U = \langle C, E \rangle$ and its firing component $Q \in \mathbf{FC}[U]$, let with the pre-set $\bullet Q$ and post-set Q^\bullet of Q , some multisets $\overline{\bullet Q}: \bullet Q \rightarrow \mathbb{N}_\omega$ and $\overline{Q^\bullet}: Q^\bullet \rightarrow \mathbb{N}_\omega$ be given as additional information. The value $\overline{\bullet Q}(x)$ is a *weight* (or *multiplicity*) of monomial $E_Q(x)$ and the value $\overline{Q^\bullet}(x)$ - a *weight* (or *multiplicity*) of monomial $C_Q(x)$. Let $\overline{\bullet Q}(x) = 0$ for $x \notin \bullet Q$ and $\overline{Q^\bullet}(x) = 0$ for $x \notin Q^\bullet$. Let cap be a total function $\mathit{cap}: \mathit{car}(U) \rightarrow \mathbb{N}_\omega \setminus \{0\}$, assigning a *capacity* to each node in the set $\mathit{car}(U)$. A c-e structure with such enhanced firing components is called a *c-e structure-with-weights of monomials and capacity of nodes*.

□

Definition 2.3. (enabled firing components and inhibitors)

For a firing component $Q \in \mathbf{FC}[U]$, let $inh[Q] = \{x \in \bullet Q : \overline{\bullet Q}(x) = \omega\}$, thus a set of nodes in the pre-set of Q , whose effect monomials $E_Q(x)$ are of weight ω . The nodes in $inh[Q]$ will play role of *inhibiting nodes* of firing component Q , as follows. For Q and state \bar{s} let us define the formula: $enabled[Q](\bar{s})$ if and only if:

$$\begin{aligned} \forall x \in inh[Q] : \bar{s}(x) &= 0 \wedge \\ \forall x \in \bullet Q \setminus inh[Q] : \overline{\bullet Q}(x) &\leq \bar{s}(x) \leq cap(x) \wedge \\ \forall x \in Q^\bullet : \overline{Q^\bullet}(x) + \bar{s}(x) &\leq cap(x) \end{aligned}$$

So, Q is enabled in the state \bar{s} iff none of inhibiting nodes $x \in \bullet Q$ contains a token and each remaining node in $\bullet Q$ does, with no fewer tokens than is the weight of its effect monomial $E_Q(x)$ and no more than capacity of each $x \in \bullet Q$. Moreover, none of $x \in Q^\bullet$ holds more tokens than their number increased by the weight of its cause monomial $C_Q(x)$, exceeds capacity of x . The inhibiting nodes of a firing components will be called its *inhibitors*. □

Definition 2.4. (Sequential and parallel semantics of extended c-e structures)

Sequential. For $Q \in \mathbf{FC}[U]$, let $[[Q]] \subseteq \mathbb{S} \times \mathbb{S}$ be a binary relation defined as: $(\bar{s}, \bar{t}) \in [[Q]]$ iff $enabled[Q](\bar{s}) \wedge \bar{t} = (\bar{s} - \overline{\bullet Q}) + \overline{Q^\bullet} \leq cap$ (Q transforms state \bar{s} into \bar{t}). *Semantics* $[[U]]$ of $U \in \mathbf{CE}$ is $[[U]] = \bigcup_{Q \in \mathbf{FC}[U]} [[Q]]$. Closure

$[[U]]^*$ and reachability and computation are defined as in Definition 1.6.

Parallel. For $G \subseteq \mathbf{FC}[U]$, $G \neq \emptyset$, containing only pairwise detached (*pwd*) firing components, let $[[G]]_{par} \subseteq \mathbb{S} \times \mathbb{S}$ be a binary relation defined as: $(\bar{s}, \bar{t}) \in [[G]]_{par}$ iff $\forall Q \in G : (enabled[Q](\bar{s}) \wedge (\bar{s} - \overline{\bullet Q}) + \overline{Q^\bullet} \leq cap)$ (say: G transforms state \bar{s} into \bar{t}). *Semantics* $[[U]]_{par}$ of $U \in \mathbf{CE}$ is: $[[U]]_{par} = \bigcup_{G \subseteq \mathbf{FC}[U] \wedge G \text{ is } pwd} [[G]]_{par}$. Closure $[[U]]_{par}^*$, reachability and computation are defined as above. □

An example of using inhibitor is in Fig.2.1. The weight of one effect (i.e. subscript) monomial $\omega \otimes z$ of the node T is ω , thus traversing arrow from T to z would require infinite number of tokens at T , which is impossible. Thus, T is the inhibiting node in firing component Q_0 and ordinary - in firing component Q_1 . In accordance with aforesaid notation of multisets, the weighted monomials are denoted by $w \otimes M$ (in grey font in figures - if $w = \omega$), where w is the weight of M - an unweighted monomial. The weight is skipped if $w = 1$. The corresponding arrows are grey and dashed in the figures if $w = \omega$.

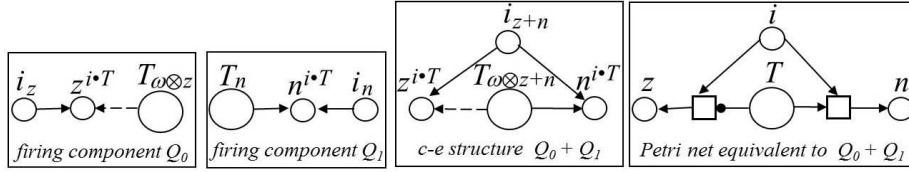


Fig.2.1. Construction of c-e structure $Q_0 + Q_1$ implementing "test zero", in accordance with Definition 1.3. A token at the node i starts testing contents of T ; on termination, the token appears at z (zero) if T is empty and at n (not zero) otherwise. The tested node T plays two roles: inhibiting in Q_0 and ordinary in Q_1 . Capacity of T is infinite, whereas of remaining nodes is 1. The empty subscript/superscript is skipped. In the Petri net counterpart, the inhibiting arrow enters the left transition.

3. Modelling of arithmetic operations in c-e structures

A natural number " n " is represented by a collection of n tokens placed at a node. Thus, the unary coding of numbers is adopted, being their basic representation in the Peano standard model of arithmetic. Although such coding is cumbersome in practical computing in comparison with codings in base greater than 1, it is used in some theoretical problems of complexity, e.g. [Gar 1978] and in data compression, e.g. [Gol 1966]. Also, such coding is used in the Church's lambda calculus [Chu 1941]. Notice that algorithms for arithmetic operations on numbers coded in base greater than 1 require referring to tables (in fact to some algorithms performed on the lower level, e.g. by computer hardware) of these operations on single digits. This is not the case with the unary coding. Such algorithms only rewrite one digit, a token, in our case, from one place to another. In this section, modelling of arithmetic operations exemplifies some capability of c-e structures and shows exact number of token moves in seven exemplary c-e structures, in order to get results of typical arithmetic operations. The modelling is presented in the form of c-e structures as graphs with nodes labelled by the identifiers, with polynomials in their superscript (causes) and subscript (effects), as exemplified in Figs 1.2 and 2.1. Nodes for input and output data of unbounded capacity ω , will be drawn as bigger circles with double edge. Remaining nodes, the "control", are of capacity 1 and drawn as smaller circles. The grey dashed arrows represent inhibiting. To distinguish multiplication of numbers from multiplication of polynomials in c-e structures, the symbol " \times " is used in case of numbers. Each c-e structure has initial node " $start$ " and final " $stop$ ", that hold a token only at the beginning and termination of its activity, respectively.

3.1. The addition (iterated adding of unity)

The sum $n + m$ is obtained by m -times adding 1 to n or n -times adding 1 to m .

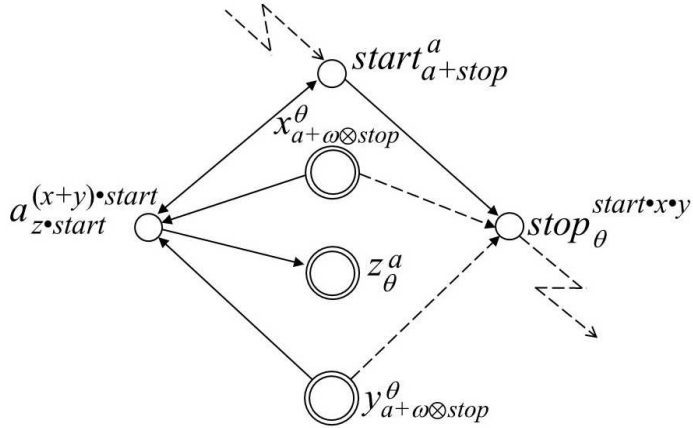


Fig.3.1 Implementation of $n + m$. Placing n tokens at x and m tokens at y and one token at $start$, makes the c-e structure terminate with $n + m$ tokens at z and one token at $stop$. The result requires $2 \times (n + m) + 1$ moves (the moves $a \rightarrow z$ and $a \rightarrow start$ take place simultaneously, so, are counted as one move)

3.2. The subtraction (iterated subtracting of unity)

The difference $n - m$ is obtained by m -times subtracting 1 simultaneously from n and m , until n becomes 0, provided that $m \leq n$.

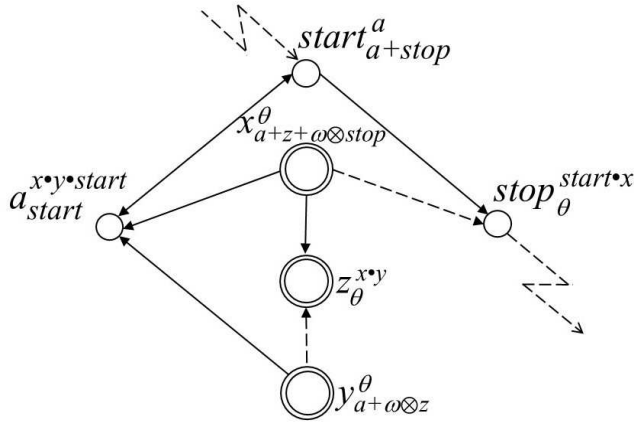


Fig.3.2 Implementation of $n \dot{-} m$. Placing n tokens at x and m tokens at y and one token at $start$, makes the c-e structure terminate with $n - m$ tokens at z if $m \leq n$ and no token at z if $m > n$ and one token at $stop$. The result requires $2 \times m + (n - m) + 1 = n + m + 1$ moves ($x \rightarrow a$ and $y \rightarrow a$ occur simultaneously, so, are counted as one move; $n \dot{-} m = \begin{cases} n - m & \text{if } m \leq n \\ 0 & \text{if } m > n \end{cases}$)

3.3. The multiplication (iterated adding of the same number)

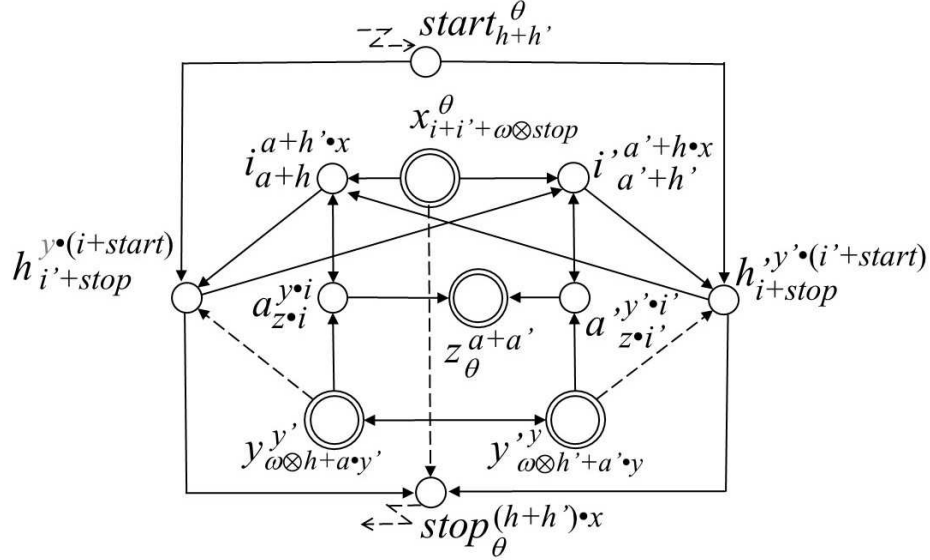


Fig.3.3 Implementation of $n \times m$. Placing n tokens at x and m tokens at y and one token at $start$, makes the c-e structure terminate with $n \times m$ tokens at z and one token at $stop$. The result requires $2 \times (n \times m + n + 1)$ moves.

Remark. The number $p = 2 \times (n \times m + n + 1)$ of moves can be easily calculated inductively. The progress of result with growing n and m is depicted in Table 3.1, where rows correspond to n and columns correspond to m . The (n, m) entry contains the pair $\langle n \times m; p \rangle$. Notice that multiplication is commutative with respect to the value of the product but not with respect to the number of moves needed to obtain the product.

Table 3.1

	1	2	3	4	5	6	7	8	9	10
1	1;6	2;8	3;10	4;12	5;14	6;16	7;18	8;20	9;22	10;24
2	2;10	4;14	6;18	8;22	10;26	12;30	14;34	16;38	18;42	20;46
3	3;14	6;20	9;26	12;32	15;38	18;44	21;50	24;56	27;62	30;68
4	4;18	8;26	12;34	16;42	20;50	24;58	28;66	32;74	36;82	40;90
5	5;22	10;32	15;42	20;52	25;62	30;72	35;82	40;92	45;102	50;112
6	6;26	12;38	18;50	24;62	30;74	36;86	42;98	48;110	54;122	60;134
7	7;30	14;44	21;58	28;72	35;86	42;100	49;114	56;128	63;142	70;156
8	8;34	16;50	24;66	32;82	40;98	48;114	56;130	64;146	72;162	80;178
9	9;38	18;56	27;74	36;92	45;110	54;128	63;146	72;164	81;182	90;200
10	10;42	20;62	30;82	40;102	50;122	60;142	70;162	80;182	90;202	100;222

3.4. The division with neglected remainder (iterated subtracting of divisor from dividend)

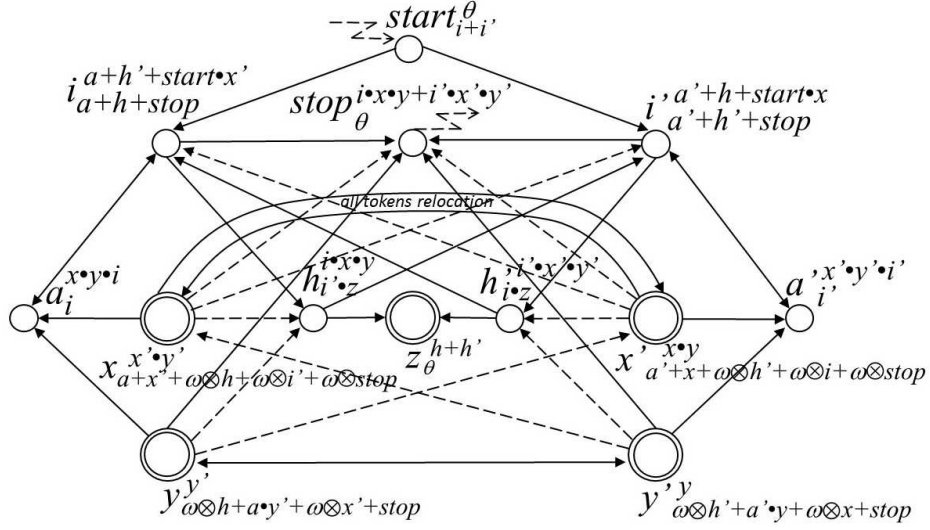


Fig.3.4 Implementation of $\lfloor \frac{n}{m} \rfloor$. Placing n tokens at x and $m > 0$ tokens at y and one token at $start$, makes the c-e structure terminate with $\lfloor \frac{n}{m} \rfloor$ tokens at z and one token at $stop$. The result requires $p = f(n, m)$ moves and this value is computed from equation $f(n, m) = n + m + 2 + f(n - m, m)$, for $m \leq n$ and $p = 2 \times (n + 1)$ for $m > n$

Remark. The number $p = f(n, m)$ of moves is obtained as solution to the recursive equation $f(n, m) = n + m + 2 + f(n - m, m)$ for $m \leq n$ and $p = 2 \times (n + 1)$ for $n < m$. Notice that if we admit $m = 0$ then the c-e structure in Fig.3.4 might run into the "livelock": the two-way relocation of tokens between nodes x and x' might never end, with permanently empty z if $n > 0$, and with unboundedly growing number of tokens at z if $n = 0$. The equation results from observation of the c-e structure's activity, exploiting its symmetric (with respect to the vertical line between $start$ and $stop$) shape. The progress of p with growing n (dividend) and m (divisor) is depicted in Table 3.2, where rows correspond to n and columns correspond to m . The (n, m) entry contains the pair $\langle \lfloor \frac{n}{m} \rfloor; p \rangle$.

3.6. The exponentiation (iterated multiplication of the number by itself)

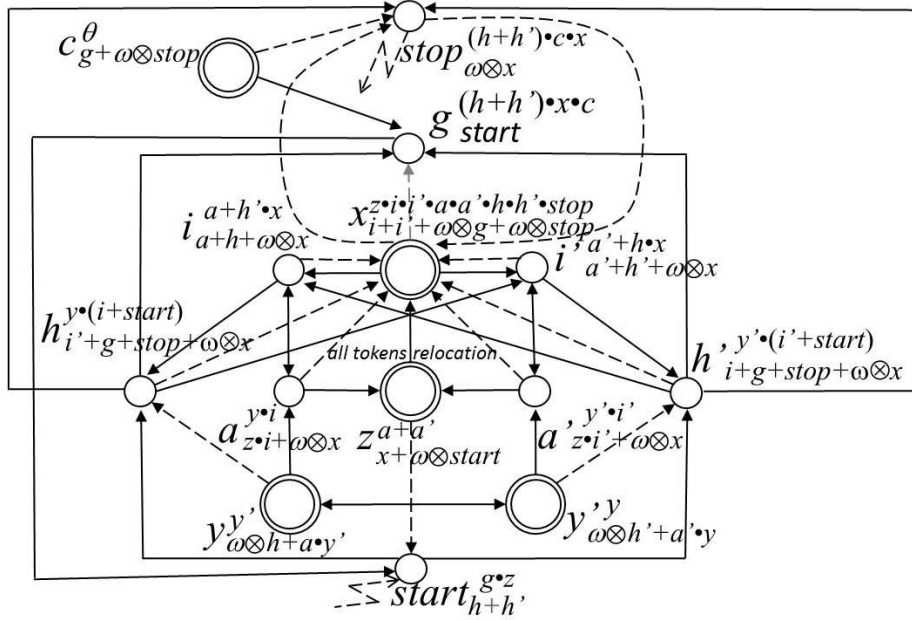


Fig.3.6 Implementation of exponentiation n^k for $k > 1$. Placing n tokens at x as well as at y , and $k - 2$ tokens at c (the counter of multiplication iterations) and one token at $start$, makes the c - e structure terminate with n^k tokens at z , and with one token at $stop$.

Remark. $0^k = 0$ and $1^k = 1$ is obtained by placing k tokens at c . The number of moves required to obtain the result n^k is straightforwardly calculated applying number p , for multiplication (Fig.3.3) with $m = n$. Notice that the value of 0^k and of 1^k does not depend of k , but the number of moves - does.

3.7. The factorial (iterated multiplication with successively decreased argument)

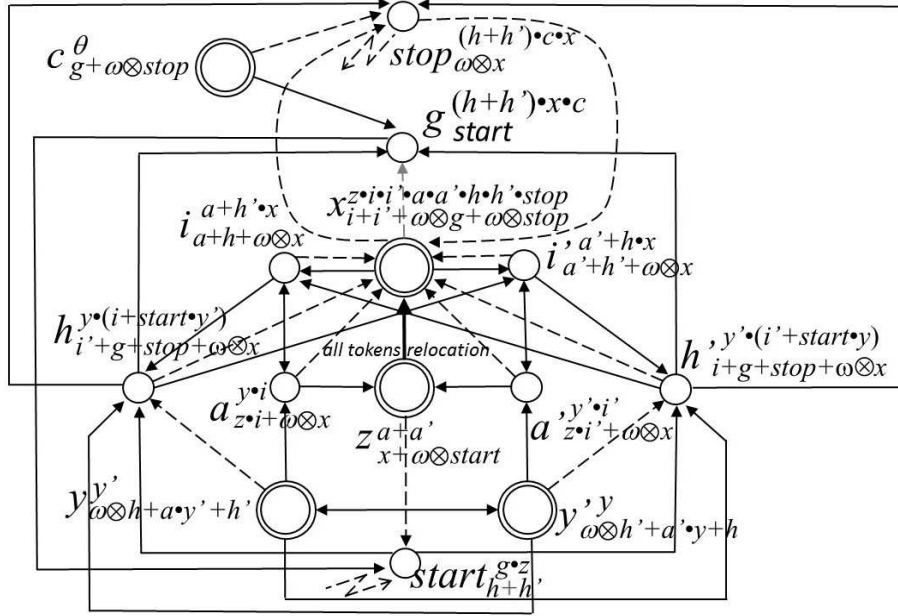


Fig.3.7 Implementation of factorial $n!$ for $n > 1$. Placing n tokens at x as well as at y , and $n - 2$ tokens at c , and one token at $start$, makes the c - e structure terminate with $n!$ tokens at z , and with one token at $stop$.

Remark. The number of moves required to obtain the result $n!$ is calculated as in case of exponentiation, but with removing one token from y and y' at any step of iteration. Notice that the following moves (i.e. firing components): $(y \rightarrow h') \bullet (start \rightarrow h')$ and $(y' \rightarrow h) \bullet (start \rightarrow h)$ remove tokens from y and from y' . Thus, they make decrease by 1 contents of y and y' at any step of iteration.

4. Computing of some arithmetic functions - examples

4.1. Summation of sequence of numbers, sequential computation

$y := x1 + x2 + x3 + x4 + x5 + x6 + x7$,
 initial state: $s(start) = 1$, $s(a) = 0$, $s(y) = 0$, $s(x1) = 3$, $s(x2) = 1$,
 $s(x3) = 2$, $s(x4) = 0$, $s(x5) = 4$, $s(x6) = 0$, $s(x7) = 1$

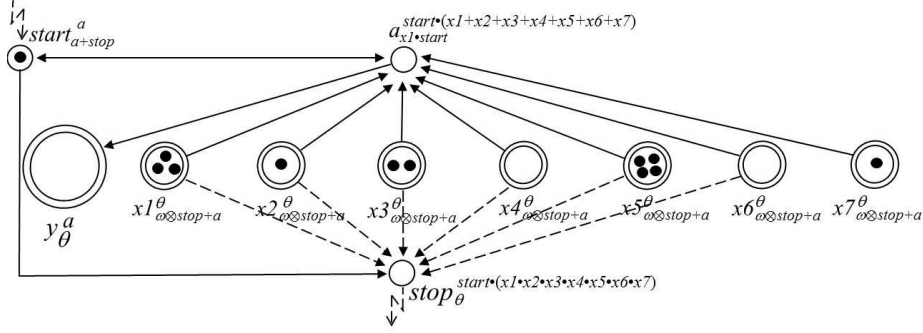


Fig.4.1 On termination, the variable y contains 11 tokens

4.2 Inner product \boxtimes of two vectors, sequential computaion

$\langle x1, x2, x3, x4, x5 \rangle \boxtimes \langle y1, y2, y3, y4, y5 \rangle$
 $z1 := x1 \times y1; z2 := x2 \times y2; z3 := x3 \times y3; z4 := x4 \times y4; z5 := x5 \times y5$
 $z1 := z1 + z2 + z3 + z4 + z5.$ Here, as in previous sections, " \times " denotes multiplication of numbers.

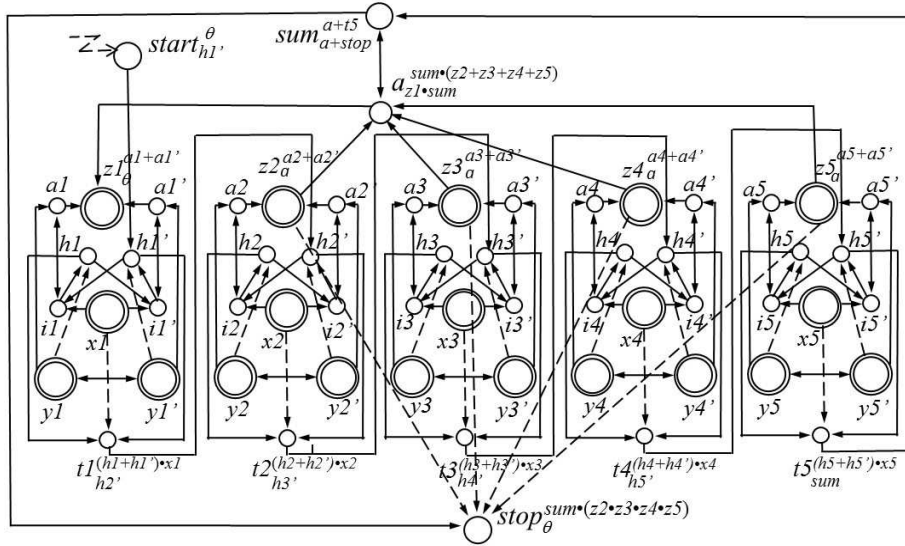


Fig.4.2 Inner roduct of vectors $\langle x1, x2, x3, x4, x5 \rangle$ and $\langle y1, y2, y3, y4, y5 \rangle$. Sequential computation. On termination, result is in the variable $z1$. In this figure, nodes with not shown superscript and subscript polynomials, are annotated as follows: $ak_{zk \cdot ik}^{yk \cdot ik}$, $ak_{zk \cdot ik'}^{yk' \cdot ik'}$, $hk_{ik' + tk}$, $h1_{i1+t1}^{start+y1' \cdot i1'}$, $hk_{ik+tk}^{yk \cdot (ik'+t[i-1])}$ if $k > 1$, $ik_{ak+hk}^{ak+hk' \cdot xk}$, $ik_{ak'+hk'}^{ak'+hk' \cdot xk}$, $xk_{ik+\omega \otimes tk}^{\theta}$, $yk_{\omega \otimes hk+ak}^{yk}$, $yk_{\omega \otimes hk'+ak'}^{yk'}$, where $k = 1, 2, 3, 4, 5$. Superscripts and subscripts of node names are readily seen from the arrows entering and exiting these nodes

4.3 Inner product \boxtimes of two vectors, parallel computaion

The same vectors as in 4.2.

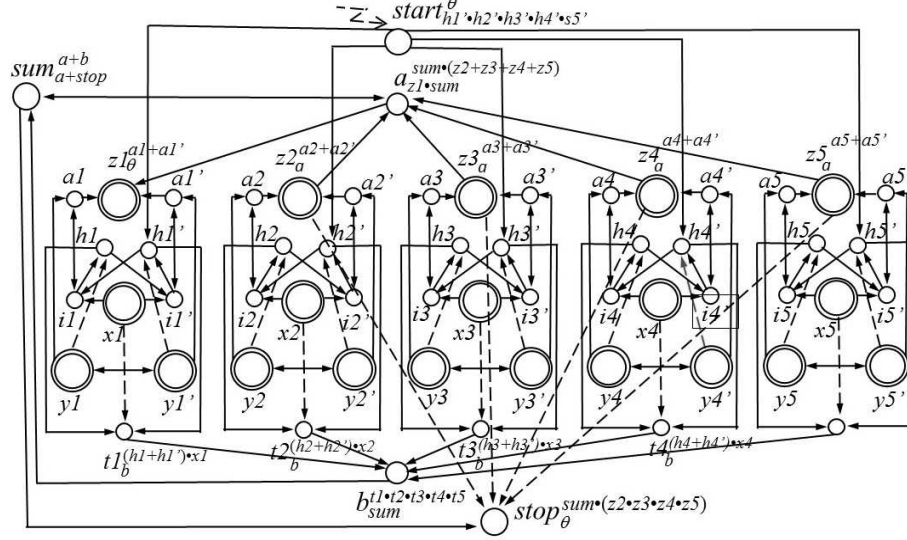


Fig.4.3 Inner roduct of vectors $\langle x_1, x_2, x_3, x_4, x_5 \rangle$ and $\langle y_1, y_2, y_3, y_4, y_5 \rangle$. Parallel computation. On termination, result is in the variable $z1$. In this figure, nodes with not shown superscript and subscript polynomials, are annotated as follows: $ak_{zk \bullet ik}^{yk \bullet ik}$, $ak_{zk \bullet ik'}^{yk' \bullet ik'}$, $hk_{ik' + tk}$, $hk_{ik + tk}^{start + yk' \bullet ik'}$, $ik_{ak + hk}^{ak + hk' \bullet xk}$, $ik_{ak' + hk'}^{ak' + hk' \bullet xk}$, $xk_{ik + ik' + \omega tk}^\theta$, $yk_{\omega \otimes hk + ak \bullet yk'}$, $yk_{\omega \otimes hk' + ak' \bullet yk}$, where $k = 1, 2, 3, 4, 5$.

Final note

It is shown in [Cza 2019] the behavioural equivalence of the c-e structures and place/transition Petri nets with inhibitors. Such systems enjoy the full Turing expressive power, thus are capable of arithmetic computations. This draft is intended to explicitly show, how the abovementioned seven typical arithmetic operations can be performed by c-e structures. Since every arithmetic function may be composed of such operations, with possible usage of recursion and the minimum combinator, the c-e structures for these operations, suitably combined, are expected to expose some details of computing such functions, in particular their exact, not only asymptotic, computational cost. In the modelling of the seven arithmetic operations, the ordinary (sequential, not parallel - cf. [Cza 2019]) semantics of c-e structures has been applied. This follows the classic standard modelling of arithmetic. Application of the parallel semantics may influence, at most, the number of simultaneous token relocations, not computing power of the modelling, i.e. the class of computed functions. Finally, notice that the iteration of actions in above c-e structures, is accomplished by oscillation of work between their symmetric parts. This corresponds to usage

of the combinator of recursion in the classic definition of computable functions, whereas the "test zero" (Fig.2.1) makes possible to accomplish the combinator of minimum.

References

[Chu 1941] Church A. *The calculi of lambda conversion*, Annals of Mathematics Studies, nr 6, Princeton University Press, 1941

[Cza 2019] Czaja L. *Cause-Effect Structures. An Algebra of Nets with Examples of Application*, Lecture Notes in Networks and Systems 45, Springer 2019

[Gar 1978] Garey M. R.; Johnson, D. S., '*Strong*' NP-completeness results: Motivation, examples, and implications, Journal of the ACM, 25 (3), 1978, pp. 499–508,

[Gol 1966] Golomb S.W., *Run-length encodings*, IEEE Transactions on Information Theory, IT-12 (3), 1966, pp. 399–401