Polynomial agorithms for BR-nets and for a fragment of Girard's Linear Logic

D.A.Archangelsky, M.I.Dekhtyar, M.A.Taitslin

Department of Computer Science ,
Tver State University, Russia
e-mail: mat@mat.tvegu.tver.su

Abstract

In this paper we consider some classes of nets with bounded types of distributed resources (BR-nets). The succsesful execution problem for BR-nets is equivalent to the provability problem for the logical calculus based on Horn fragment of Girard's Linear Logic [2]. We show that, in spite of NP-completeness of the problem for all BR-nets, it is in PTIME for some interesting subclasses of BR-nets.

1.Introduction

The idea of relating the concurrent computations to the linear logic proofs was first proposed by Girard [6] and then considered in numbers of works (see [4, 8]). BR-nets were introduced to represent concurrency in distributed systems of bounded resource transformations. Every node of a BR-net contains a multiset of resources of fixed types called supplies and rules of supply transformations (converters). A request for a BR-net consists of a multiset of source resources, a multiset of resulting resources and a goal node where the result should be obtained. An execution of a BR-net for a given multiset of requests includes two stages. The first stage is an initial resource distribution among the nodes of the net between concurrent requests. The second stage consists of applying converters to supplies at the nodes and of transmitting resources between connected nodes. Each supply and each converter can be used only once and all the distributed resources should be used.

In [2] a linear logic calculus was constructed that adequately describes the behavior of BR-nets. It is based upon Horn fragment of Multiplicative Linear Logic (it is in fact its conservative extention). Behavior of BR-nets and proofs in corresponding calculus catch following significant features:

(i) simultaneous execution of concurrent requests (or simultenuous proof of a multiset sequents),

(ii) use of a restricted amount of initially given resources (axioms).

It seems that in previous studies of Linear Logic these features were not completely reflected.

In this paper we consider the successful execution problem for BR-nets which is equivalent to the provability problem for the logical calculus introduced in [2]. This problem is NP-complete in the class of all BR-nets (in the case of one-node nets with unrestricted converters set it follows from [9], some other cases are given by theorems 1 and 3(iii) bellow). We examine three kinds of restrictions on BR-nets. The first is a fixed number of resource types. The second is a tree-like graph structure of BR-nets, and the third is an acyclicity of the convertors dependency graph (i.e. any supply can not be used to produce itself). The first two conditions provide a polynomial time algorithm for the problem (theorem 2(ii)). This result improves a subexponential algorithm obtained in [1] for one-node nets and one request. The last condition was used in [3] to obtain the convertors sets with the maximal degree of concurrency. Here we show that with the second one it allows to obtain polynomial time algorithms for one request or for one-node nets, but in the case of BR-nets with two nodes and multiple requests the problem is NP-complete (theorem 3).

2. BR-nets

We fix a finite set $\mathbf{S}$. The elements of $\mathbf{S}$ will be called *supplies*. The set $\mathbf{S}List$ of supply lists is the least satisfying the following conditions: $\mathbf{S} \subset \mathbf{S}List$; if $A, B \in \mathbf{S}List$, then $(AB) \in \mathbf{S}List$. Two lists $A, B \in \mathbf{S}List$ are equivalent iff the multisets of elements of $\mathbf{S}$ included in A and B are equal. For example, for $\mathbf{S} = \{a, b, c\}$, $((aa)(bc))$ is equivalent to $(a(b(ca)))$ but is not equivalent to $(a(bc))$. We do not distinct a list $A \in \mathbf{S}List$ and the multiset of elements of $\mathbf{S}$ included in A. An exponential notation will be used for representing multisets. For example a^2b^3c will correspond to a multiset $\{a, a, b, b, b, c\}$ and to any list constructed of these and only these letters.

Let X and Y be supply lists. Then an expression $(X \to Y)$ is called *a converter*. Converters $(X \to Y)$ and $(U \to V)$ are equivalent if X is equivalent to U and Y is equivalent to V. We fix such a finite set $\mathbf{P}$ of converters that any two different converters from $\mathbf{P}$ are not equivalent. The elements of $\mathbf{P}$ will be called basic converters. Let $\mathbf{R} = \mathbf{S} \cup \mathbf{P}$. $\mathbf{R}$ is said to be *a set of basic resources.* The definition of $\mathbf{R}List$ is similar to the definition of $\mathbf{S}List$. The set $\mathbf{R}List$ is the least satisfying the following conditions: $\mathbf{R} \subset \mathbf{R}List$; if $A, B \in \mathbf{R}List$, then $(AB) \in \mathbf{R}List$. Let $[A]r$ be the number of copies of element $r \in \mathbf{R}$ in multiset (list) $A \in \mathbf{R}List$. This means that $[r]r = 1$, $[s]r = 0$ for different elements s, r from $\mathbf{R}$ and $[(AB)]r = [A]r + [B]r$. For example

$$[((aa)(a \to (ba))(bc))]a = [a^2bc(a \to (ba))]a = 2.$$

A BR-net $\mathcal{M}$ over $\mathbf{R}$ is a pair $\langle G, f\rangle$ where $G = (V, E)$ is a finite directed graph and $f : V \times \mathbf{R} \Rightarrow \mathbf{N}$ is a function which for every node $\alpha \in V$ and for each kind of resources $r \in \mathbf{R}$ defines the amount $f(\alpha, r)$ of resource r stored in the node α.

A triplet $\langle A, B, \alpha\rangle$ where A and B are mulisets of elements of $\mathbf{R}$, and $\alpha \in V$ is called *a request.* We call A *a source* of the request, B is *a result* of the request and α is *a goal node* of the request.

Consider a BR-net $\mathcal{M} = \langle (V, E), f\rangle$ and a multiset of requests

$$\mathcal{Q} = \{Q_1, \ldots, Q_n\}, \text{ where } Q_i = \langle A_i, B_i, \alpha_i\rangle \text{ for } i = 1, \ldots, n.$$

An initialization of $\mathcal{M}$ for $\mathcal{Q}$ consists of following:

(i) for every request Q_i a number $k = k(i)$ of subrequests is defined and Q_i is divided on k subrequests

$$Q_i^{(1)} = \langle A_i^{(1)}, B_i^{(1)}, \alpha_i\rangle, \ldots, Q_i^{(k)} = \langle A_i^{(k)}, B_i^{(k)}, \alpha_i\rangle$$

such that

$$A_i = A_i^{(1)} \cup \ldots \cup A_i^{(k)}, \quad B_i = B_i^{(1)} \cup \ldots \cup B_i^{(k)};$$

(ii) in every node $\alpha \in V$ for every $1 \le i \le n$ and every $1 \le j \le k(i)$ a certain amount of resources $\mathbf{R}(\alpha, i, j)$ reserved for subrequest $Q_i^{(j)}$.

Let

$$f_{\text{init}} : (V \times \mathbf{R} \times \mathbf{N} \times \mathbf{N}) \Rightarrow \mathbf{N}$$

be a function that defines an initial distribution of resources: for every $1 \le i \le n$ and every $1 \le j \le k(i)$, $f_{\text{init}}(\alpha, r, i, j)$ is equal to the amount (the number of copies) of resource r initially reserved in node α for subrequest $Q_i^{(j)}$. Then at initial moment for every $r \in \mathbf{R}$ the multiset $\mathbf{R}(\alpha, i, j)$ will contain $f_{\text{init}}(\alpha, r, i, j)$ copies of r.

The following conditions should hold for an initial distribution :

(i) sufficiency of resources in each node :

$$(\forall \alpha \in V)(\forall r \in \mathbf{R})(\sum_{i=1}^{n}\sum_{j=1}^{k(i)} f_{\text{init}}(\alpha, r, i, j) \le f(\alpha, r));$$

(ii) completeness of resource reservation for each subrequest :

$$(\forall i \in [1, n])(\forall j \in [1, k(i)])(\forall r \in \mathbf{R})(\sum_{\alpha \in V} f_{\text{init}}(\alpha, r, i, j) = [A_i^{(j)}]r).$$

An execution of the BR-net $\mathcal{M}$ on a multiset of requests $\mathcal{Q}$ with an initialization determined by f_{init} is a sequence $\sigma = s_1 s_2 \ldots$ of steps which affect some of the multisets $\mathbf{R}(\alpha, i, j)$. Execution steps can be of three different types:

(a)*Resource transformation at a node.*

If for some $\alpha \in V$ and $i \in [1, n], j \in [1, k(i)]$, $\mathbf{R}(\alpha, i, j)$ is equivalent to $(C(A(A \to B)))$ then the step consists of changing $\mathbf{R}(\alpha, i, j)$ to (CB), i.e. of application of the converter $(A \to B)$ to the supply list A and obtaining the supply list B as a result. This step is active for these α, i.

(b)*Supply transmission between nodes.*

If $(\alpha, \beta) \in E$, $\mathbf{R}(\alpha, i, j) = A$ and $A \in \mathbf{S}List$ (A does not contain converters) then the step consists of changing $\mathbf{R}(\alpha, i, j)$ to an empty set and adding A to $\mathbf{R}(\beta, i, j)$. This step is active for these β, i.

(c)*Subrequests union.*

If for some $\alpha \in V$, $i \in [1, n]$, $j, j\prime \in [1, k(i)]$, $\mathbf{R}(\alpha, i, j)$ is not empty and $j\prime \ne j$ then the step consists of adding $\mathbf{R}(\alpha, i, j)$ to $\mathbf{R}(\alpha, i, j\prime)$ and of changing $\mathbf{R}(\alpha, i, j)$ to an empty set (i.e. subrequests $Q_i^{(j)}$ and $Q_i^{(j\prime)}$ are being united in the node α). This step is active for these α, i.

An execution $\sigma = s_1 s_2 \ldots s_m$ of $\mathcal{M}$ on $\mathcal{Q}$ with f_{init} is called *successful* iff after the step s_m the following two conditions hold:

(i) $(\forall i \in [1,n])(\forall j \in [1,k(i)])(\mathbf{R}(\alpha_i, i, j) = B_i^{(j)})$,

and

(ii) $(\forall i \in [1,n])(\forall j \in [1,k(i)])(\forall \beta \neq \alpha_i)(\mathbf{R}(\beta, i, j) = \emptyset)$.

A multiset $\mathcal{Q}$ of requests *succeeds* on BR-net $\mathcal{M}$ iff there exist an initialization f_{init} and a successful execution σ of $\mathcal{M}$ on $\mathcal{Q}$ with f_{init}.

Example 1. Consider the BR-net $\mathcal{M}_1$ shown on Figure 1.

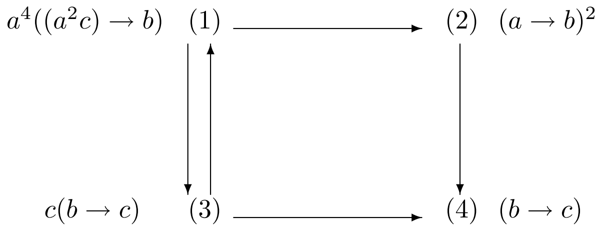

Figure 1.

It has four nodes : 1, 2, 3, 4. The set of basic resources consists of supplies $\{a, b, c\}$ and converters $\{((a^2c) \to b), (a \to b), (b \to c)\}$. Resource presence is shown near the nodes. For example, $f(1, a) = 4$, $f(2, (a \to b)) = 2$. Consider a multiset $\mathcal{Q}$ of two requests
$\{\langle ca^3((a^2c) \to b)(a \to b)(b \to c)^2, c^2, 4\rangle, \langle a(a \to b), b, 2\rangle\}$.

Define the initial distribution of resources:

$$k(1) = 2, k(2) = 1,$$

$$\mathbf{R}(1,1,1) = a^2((a^2c) \to b), \mathbf{R}(1,1,2) = a, \mathbf{R}(1,2,1) = a,$$

$$\mathbf{R}(2,1,2) = \mathbf{R}(2,2,1) = (a \to b),$$

$$\mathbf{R}(3,1,1) = c, \mathbf{R}(3,1,2) = (b \to c), \mathbf{R}(4,1,2) = (b \to c).$$

Then it is easy to define such a sequence of steps that will lead to the following distribution:

$$\mathbf{R}(2,2,1) = b, \mathbf{R}(4,1,1) = c, \mathbf{R}(4,1,2) = c$$

(all the other resource boxes will be empty). Therefore the set of requests $\mathcal{Q}$ succeeds on BR-net $\mathcal{M}_1$.

We will consider BR-nets satisfying some restrictions. We said that a BR-net $\mathcal{M}$ is *a tree-like* if its graph $G = (V, E)$ is a tree. For tree-like BR-nets the definition of requests execution can be simplified.

Lemma 1 *Suppose $\mathcal{M}$ is a tree-like BR-net.*

(i). If the multiset $\{Q_1, \ldots, Q_n\}$ of requests succeeds on $\mathcal{M}$ then there is such a successful execution with such an initialization that no request is divided on subrequests. So $k(i) = 1$ for $i \in [1, n]$.

(ii). If the multiset of requests succeeds on the $\mathcal{M}$ with such an initialization that no request is divided on subrequests then there is such a successful execution $\sigma = s_1 s_2 \ldots s_l$ with the same initialization that for $0 < p < q \leq l$, if s_p is active for α, i and s_q is active for β, j then the tree level of α is less than or equal to the tree level of β.

This lemma shows that for the class of trees BR-nets are equivalent to RT-nets introduced in [5].

To define another restriction we associate with a set $\mathbf{P}$ of basic converters an oriented graph $G(\mathbf{P}) = (\mathbf{S}, E_P)$. The set $\mathbf{S}$ of nodes consists of all supplies. An edge $(x, y) \in E_P$ iff there exists a converter $(X \to Y) \in \mathbf{P}$ such that x is contained in X and y is contained in Y. The property we are interesting in is *an acyclicity* of $G(\mathbf{P})$. It means that there does not exist a sequence of resource transformations in which some supply a is used to produce another instance of a. It is shown in [3] that if $G(\mathbf{P}$ is acyclic, then convertors can be applied in any order to obtain successful computation in one node. The next example shows that it is not the case for the BR-nets with nontrivial graphs.

Example 2.

Consider the BR-net $\mathcal{M}_2$ shown on Figure 2.

$$\begin{tikzcd}[row sep=large]
a(a \to b) \quad (1) \arrow[d] \\
d(d \to a)(a \to c) \quad (2) \arrow[d] \\
a \quad (3)
\end{tikzcd}$$

Figure 2.

It has nodes : 1, 2, 3. The set of basic resources consists of supplies $\{a, b, c, d\}$ and converters $\mathbf{P} = \{(a \to b), (d \to a), (a \to c)\}$. Resource presence is shown near the nodes. It is easy to see that $\mathcal{M}_2$ is a tree-like BR-net and that the graph $G(\mathbf{P})$ is acyclic.

Consider a multiset $\mathcal{Q}$ of two requests

$$\{\langle ad(a \to b)(d \to a), ab, 3\rangle, \langle a(a \to c), c, 3\rangle\}.$$

and request

$$\langle aad(a \to b)(d \to a)(a \to c), abc, 3\rangle.$$

It is easy to see that this request succeeds on $\mathcal{M}_2$ and each request from $\mathcal{Q}$ succeeds itself on $\mathcal{M}_2$. But the multiset does not succeed on $\mathcal{M}_2$ because we cannot unite results of distinct requests.

3. Complexity of successeful execution problem

In this section we consider the next *successful execution problem*: given BR-net $\mathcal{M}$ and a multiset of requests $\mathcal{Q}$ does $\mathcal{Q}$ succedes on $\mathcal{M}$? For one-node BR-nets the question on complexity of this problem is equivalent to the question on complexity of solvability problem of Horn fragment of Multiplicative Linear Logic, which was raised in [10].It was shown in [9] that solvability problem for this fragment is NP-complete. In [1] NP-complete problem 3-PARTITION (see [7]) is being reduced to the provability problem of Horn sequents with 2 letters.

It is easy to see that successful execution problem is in NP for the class of all BR-nets. It is NP-hard for the 1 letter alphabet of supplies and an one element fixed set of converters.

Theorem 1 *Let the set of resource* $\mathbf{R}$ *be* $\{x, (x^3 \to x)\}$. *Then the problem "Does the request* $\langle A, B, \alpha\rangle$ *succeed on the BR-net* $\mathcal{M}$ *over* $\mathbf{R}$*?" is NP-complete.*

Sketch of the proof. Let $I = (a_1, ..., a_{3m})$ be an instance of 3-PARTITION problem. Let

$$a_1 + \ldots + a_{3m} = ma.$$

Consider BR-net $\mathcal{M} = \langle (V, E), f\rangle$ where

$$V = \{i \mid 1 \le i \le 3m\} \bigcup \{(i, j, k) \mid 1 \le i < j < k \le 3m \text{ and } a_i + a_j + a_k = a\} \bigcup \{g\},$$

$$E = \{(i, (i, j, k)), (j, (i, j, k)), (k, (i, j, k)), ((i, j, k), g) \mid (i, j, k) \in V\};$$

$f(i, x) = 1$ for all the nodes $i \in [1, 3m]$, $f((i, j, k), (x^3 \to x)) = 1$ for all the nodes of a form (i, j, k), and $f(v, r) = 0$ for any other pair of arguments. Then the request $(x^{3m}(x^3 \to x)^m, x^m, g)$ succeeds on $\mathcal{M}$ iff $I \in$3-PARTITION.

BR-net $\mathcal{M}$ constructed in the proof above has acyclic graph. But it is not a tree-like BR-net. The next theorem shows that for tree-like BR-nets the problem is polynomial.

Theorem 2 *Fix numbers* k *and* m. *Let* $\mathcal{M}$ *be a tree-like BR-net.*

(i). Let set $\mathbf{P}$ *of converters contains less than* k *elements. Then the problem "Does the multiset of* m *requests*

$$\mathcal{Q} = \{\langle A_1, B_1, \alpha_1\rangle, \ldots, \langle A_m, B_m, \alpha_m\rangle\}$$

succeed on the BR-net $\mathcal{M}$ *over* $\mathbf{R}$*?" can be solved in time* $O(n^{\log(n)})$, *where* n *is the size of the input.*

(ii). Let the set $\mathbf{R}$ *of resources contains less than* k *elements. Then the problem "Does the multiset of* m *requests*

$$\mathcal{Q} = \{\langle A_1, B_1, \alpha_1\rangle, \ldots, \langle A_m, B_m, \alpha_m\rangle\}$$

succeed on the BR-net $\mathcal{M}$ *over* $\mathbf{R}$*?" is in PTIME.*

Sketch of the proof (i). The algorithm description is analogous to one that was given in [1]. We can suppose that $B_1, \ldots, B_m$ are multisets of supplies. A request $\langle (CD), B, \alpha\rangle$ where C is a multiset of supplies and D is a multiset of converters is correct if B is obtained from C by deleting the premises of all the converters from D and adding the conclusions of all the converters. At first we check the correctness of the requests. If a request from $\mathcal{Q}$ is incorrect then $\mathcal{Q}$ does not succeed on $\mathcal{M}$.

Let all the requests be correct.

First we consider the case when for each $i = 1, \ldots, m$, the multiset A_i contains less than two converters. In this case, if A_1 contains a converter, we choose the nearest to α_1 node β located on the path from the root to α_1 from such nodes that the resources stored in the node contain the converter. Then we initially distribute the converter to the node β. The supplies needed for the converter are chosen from the resources stored in the nearest to β nodes located on the path from the root to β, and the supplies are distributed to the nodes. The additional supplies are chosen from the resources stored in the nearest to α nodes located on the path from the root to α. We distribute A_1 and then we change $\mathcal{M}$ by deleting the resources initially reserved for A_1. After then we are going to consider $A_2, \ldots, A_m$, if $m > 1$.

We attempt to present A_i as (U_iV_i) in all possible ways such that multiset U_i contains the one-half of all the converter members of A_i. For each such presentation we choose a node β_i located on the path from the root to α_i. We attempt to satisfy

$$\{\langle U_1, W_1, \beta_1\rangle, \ldots, \langle U_m, W_m, \beta_m\rangle\}$$

for some multisets $W_1, \ldots, W_m$ of supplies. If the attempt is successful we change $\mathcal{M}$ by deleting the resources initially reserved in node β_i for subrequest $\langle U_i, W_i, \beta_i\rangle$ from the resources stored in the node β_i and adding W_i to the resources stored in the node β_i for all i. Then for multiset

$$\{\langle (V_1W_1), B_1, \alpha_1\rangle, \ldots, \langle (V_mW_m), B_m, \alpha_m\rangle\}$$

of requests, we attempt to find a successful execution with such an initialization that $\mathbf{R}(\gamma, i, j)$ is empty if γ is located on the path from the root to β_i and is distinct from β_i.

(ii). We improve the proof construction algorithm from [1]. We can suppose that $B_1, \ldots, B_m$ are multisets of supplies. At first we check the correctness of the requests. If a request from $\mathcal{Q}$ is incorrect then $\mathcal{Q}$ does not succeed on $\mathcal{M}$.

Let all the requests be correct.

A cut is such a m-tuple $(\beta_1, \ldots, \beta_m)$ of nodes that for $i = 1, \ldots, m$, the node β_i is located on the path from the root of G to α_i.

A distribution g for a cut $(\beta_1, \ldots, \beta_m)$ is a mapping which for every $i = 1, \ldots, m$ and for each kind of resources $r \in \mathbf{R}$ defines a number $g(\beta_i, r, i)$ of copies of resource r involved in the node β_i in the request of number i. The distribution is correct if for each resource r and each node α from the cut, the total number of copies of resource r involved in the node α in all the requests is less than or equal to the number $f(\alpha, r)$ of copies of r stored in α.

A state is such a pair that the first element is a correct distribution g for a cut $(\beta_1, \ldots, \beta_m)$, and the second element is a sequence

$$(U_1, W_1, \beta_1), \ldots, (U_m, W_m, \beta_m)$$

of requests where $W_1, \ldots, W_m$ are multisets of supplies. The state is correct if the following property holds:

there is a successful execution of $\mathcal{M}$ on

$$(U_1, W_1, \beta_1), \ldots, (U_m, W_m, \beta_m)$$

with such an initial distribution h that for every $i = 1, \ldots, m$ and for each kind of resources $r \in \mathbf{R}$, the number $k(i)$ of all the subrequests for the request of number i is equal to 1 and $h(\beta_i, r, i, 1) = g(\beta_i, r, i)$.

The weight of the state is the number

$$(\sum_{r \in \mathbf{R}} \sum_{i=1}^{m} [U_i]r).$$

A next state for a given correct state

$$(g, ((U_1, W_1, \beta_1), \dots, (U_m, W_m, \beta_m)))$$

is such a correct state

$$(g_1, ((C_1, D_1, \gamma_1), \dots, (C_m, D_m, \gamma_m)))$$

that there exist such a natural number j and such a resource s that the following conditions (1)–(7) hold:

(1) $0 < j < m + 1$;
(2) $s \in \mathbf{R}$;
(3) either $s \in \mathbf{S}$ or it is possible to apply s to W_j;
(4) β_j is located on the path from the root to γ_j;
(5) for $i = 1, \dots, m$, either β_i is γ_j or γ_j is not located on the path from the root to β_i;
(6) either
(6a) γ_j is β_j and

$$g_1(\gamma_j, s, j) = g(\gamma_j, s, j) + 1, \quad C_j = (sU_j),$$
$$g_1(\gamma_j, r, j) = g(\beta_j, r, j)$$

for all $r \in \mathbf{R},\ r \neq s$
or
(6b) γ_j is distinct from β_j,

$$g_1(\gamma_j, r, j) = 0$$

for all $r \in \mathbf{R}, r \neq s$,
and either
(6ba)

$$g_1(\gamma_j, s, j) = 1$$

and

$$C_j = (sU_j)$$

or
(6bb)

$$g_1(\gamma_j, s, j) = 0$$

and

$$C_j = U_j;$$

(7) for all $r \in \mathbf{R}$ and all $i = 1, \dots, m$,

$$\gamma_i = \beta_i, \quad C_i = U_i,$$
$$g_1(\gamma_i, r, i) = g(\beta_i, r, i)$$

if i is distinct from j.

The algorithm consists of steps.

Step 0. We find all the correct state with the weight 0.

Step $i + 1$. For each correct state

$$(g, ((U_1, W_1, \beta_1), \dots, (U_m, W_m, \beta_m)))$$

with the weight i we find all its next states with the weight $i + 1$. So for given $s \in \mathbf{P}$ and j we check whether it is possible to apply s to W_j. For these applicable s and for all $s \in \mathbf{S}$ we try to find such a node γ_j that β_j is located on the path from the root to γ_j, the condition (5) holds, and

$$f(\gamma_j, s) > \sum_{\beta_i = \gamma_j} g(\beta_i, s, i).$$

We call each state received a state of the level 1. For $l = 1, \dots, m$ and for each state of the level l, we find for this state all the next states of weight $i + 1$ and call them the states of level $l + 1$. As the result of the step $i + 1$ we output the union of all the states of levels $1, \dots, m + 1$.

Let the size of the input of the algorithm be n. The total number of possible states does not exceed the number of all the distributions multiplied by the number of all the sequences of m requests. So this number is less than n^{2km+m}. Checking whether one state is the next for another needs time cn for a constant c. Then the time required for the step i is less than $c(m+1)n^{4km+2m+1}$ and the time complexity of the algorithm is bounded by $c(m + 1)n^{4km+2m+2}$.

Correctness of the algorithm follows immediately from the following claim.

Claim 1 *After step i of the algorithm we obtain all the correct states with the weight i.*

The claim is proved by induction on l.

To finish the theorem proof we observe that given multiset of requests succeeds if and only if this multiset is the second element of a correct state.

If the net has the only node we don't need to use distributions. In this case the number of the supplies does not matter. So we can only suppose that the set of converters contains less than k elements. Thus theorem 3(ii) improves the theorem 6 in [1].

Now we consider the class tree-like BR-nets with acyclic graph of converters $G(\mathbf{P})$. For this class BR-nets successful execution problem is simple for one request but is NP-complete for multisets of requests.

Note that in the proof of theorem 1 we can replace $\mathbf{R}$ by $\{x, y, (x^3 \to y)\}$. So for fixed $\mathbf{R}$ with acyclic graph $G(\mathbf{P})$ the successful execution problem for one request is NP-complete. The tree-like restriction helps in this case.

Theorem 3 *Let $\mathcal{M}$ be a tree-like BR-net. Suppose graph $G(\mathbf{P})$ has no cycle. (i) The problem "Does the request*

$$\langle A, B, \alpha\rangle$$

succeed on $\mathcal{M}$ over $\mathbf{R}$?" is in PTIME.

(ii) If $\mathcal{M}$ is one-node net (i.e. $|V| = 1$), then the problem "Does the multiset of requests

$$\mathcal{Q} = \{\langle A_1, B_1, \alpha_1\rangle, \ldots, \langle A_n, B_n, \alpha_n\rangle\}$$

succeed on the BR-net $\mathcal{M}$ over $\mathbf{R}$?" is in PTIME.

(iii) For $\mathbf{S} = \{x, y, z\}$ and $\mathcal{M}$, consisting of two nodes, the problem "Does the multiset of requests

$$\mathcal{Q} = \{\langle A_1, B_1, \alpha_1\rangle, \ldots, \langle A_n, B_n, \alpha_n\rangle\}$$

succeed on the BR-net $\mathcal{M}$ over $\mathbf{R}$?" is NP-complete.

Sketch of the proof (i), (ii). See [3].

(iii). Let $I = (a_1, ..., a_{3m})$ be an instance of 3-PARTITION problem. Let

$$a_1 + \ldots + a_{3m} = ma.$$

Suppose $a/4 < a_i < a/2$ for $i = 1, \ldots, 3m$. Consider the BR-net $\mathcal{M}$ shown on Figure 3.

$$\begin{tikzcd} x^{am}(y^a \to z^a)^m(x^{a_1} \to y^{a_1}) \ldots (x^{a_{3m}} \to y^{a_{3m}}) & (1) \arrow[d] \\ x^{am(m-1)}((x^{a_1} \to y^{a_1}) \ldots (x^{a_{3m}} \to y^{a_{3m}}))^{(m-1)} & (2) \end{tikzcd}$$

Figure 3.

It has two nodes: 1, 2. The set of basic resources consists of supplies $\{x, y, z\}$ and converters. Resource presence is shown near the nodes. The graph $G = (V, E)$ is a tree.

Consider a multiset $\mathcal{Q}$ of m identical requests

$$\langle x^{am}(y^a \to z^a)(x^{a_1} \to y^{a_1}) \ldots (x^{a_{3m}} \to y^{a_{3m}}), z^a y^{a(m-1)}, 2\rangle.$$

The multiset $\mathcal{Q}$ succeeds on $\mathcal{M}$ iff $I \in$3-PARTITION.

Open problem 1. Suppose graph $G(\mathbf{P})$ has no cycle. Suppose graph $G = (V, E)$ is a tree. Fix a natural number $m > 0$. For example, let $m = 2$ and G is a line with two nodes.

Is the problem "Does the multiset of m requests

$$\mathcal{Q} = \{\langle A_1, B_1, \alpha_1\rangle, \ldots, \langle A_m, B_m, \alpha_m\rangle\}$$

succeed on the BR-net $\mathcal{M}$ over $\mathbf{R}$?" in PTIME ?

Open problem 2. Fix a list $\mathbf{P}$ of converters. Suppose graph $G = (V, E)$ is a tree. For example, let graph $G(\mathbf{P})$ has no cycle and G is a line with two nodes.

Is the problem "Does the multiset of requests

$$\mathcal{Q} = \{\langle A_1, B_1, \alpha_1\rangle, \ldots, \langle A_m, B_m, \alpha_m\rangle\}$$

succeed on the BR-net $\mathcal{M}$ over $\mathbf{R}$?" in PTIME?

Acknowledgements

This work was sponsored by International Scientific Foundation (NYF000).

References

1. Archangelsky D.A., Taitslin M.A., *Linear Logic With Fixed Resources.* Annals of Pure and Applied Logic, v. 67 (1994),pp. 3-28.
2. Archangelsky D.A., Dekhtyar M.I., Taitslin M.A., *Linear Logic For Nets With Bounded Resources.* submitted to Annals of Pure and Applied Logic.
3. Archangelsky D.A., Dekhtyar M.I., Kruglov E., Musikaev I.Kh., and Taitslin M.A. *Concurrency problem for Horn fragment of Girard's Linear Logic.* Logical Foundation of Computer Science, St.Petersburg'94, Lecture Notes in Computer Science, N 813, 1994, pp. 18–22.
4. Brown C. *Relating Petri Nets to Formulae of Linear Logic.* Technical Report ECS LFCS 89-87, Univ. of Edinburgh, 1989.
5. Dekhtyar A.M., *Multiplicative Linear Logic For Resource Transformaiton Nets.* Logical Foundation of Computer Science, St.Petersburg'94, Lecture Notes in Computer Science, N 813, 1994.
6. Girard J.Y. *Linear Logic.* Theoretical Computer Science, v. 50 (1987),pp. 1-102.
7. Garey M.R., Johnson D.S. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* W.H.Freeman and Company, San Francisco, 1979.
8. Gehlot V., Gutner C. *A Proof-theoretic Operational Semantics for True Concurrency.* Proc. International Conf. on Applic. and Theory of Petri Nets, Bonn, 1989.
9. Kanovich M.I. *Horn Programming in Linear Logic is NP-complete.* Proc.7-th Annual IEEE Symposium on Logic in Computer Science, 1992, pp. 200–210.
10.Lincoln P., Mitchell J., Scerdov A., and Shankar N. *Decision Problems for Propositional Linear Logic.* Proc.31-th IEEE Symposium on Foundation of Computer Science, 1990, pp. 662–671.