Model-Based Analysis (MBA) - Homework 1

Šimon Stupinský - xstupi00@stud.fit.vutbr.cz

Obtained points:

1. TASK

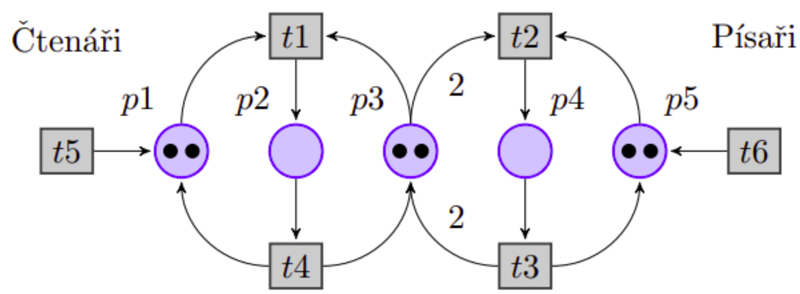

Figure 1. Readers wait in the place `p1` and read in the `p2`. Writers wait in the place `p5` and write in the `p4`.

Assignment: Assume the *P/T Petri net* from Figure 1.

(a) Design the *Reachability Tree* and with using it determine and verify the given tasks.

(b) Assume the given *P/T Petri net* without the transitions `t5`, `t6` and with initial marking $M_O = (3,0,2,0,3)$. Compute *P-Invariants* and with using it determine and verify the given tasks.

(c) Assume the given *P/T Petri net* without the transitions `t5`, `t6` and with initial marking $M_O = (3,0,2,0,3)$. Compute *T-invariants* and with using it determine and verify the given tasks.

---

a) Design the *Reachability Tree* and with using it determine and verify the given tasks.

The designed **Reachability (Coverability) Tree** is possible to see in Figure 2. When a newly generated marking repeats a marking to the path to the root, then we represent this node as a duplicated. Duplicated nodes are represented with the grey background and bold lines around. For simplicity, we show nodes with the same markings as one node with the more inputs arrows (transitions). In the case, when two or more nodes generate the same marking, then we will further expand this node from the first generation, and all next one will be marked as duplicated, and the names of these transitions are represented by the bold font. During the construction of the tree, when a new marking is greater than another marking on the path to the root, we replace larger place values with *****, what is the replacement for ω. Each lead node is represented with the colour background and we emphasise, that the same colour between the different nodes has not affiliated. The coloured nodes and transitions were used only for better orientation in the constructed tree.

Safeness. The safeness of the Petri net guarantees, that the number of tokens in each place does not exceed a value equal to one and it is the special case of the more general property called by boundedness. The Petri net is safe, when the each its place is safe, whereas is applied, that the place is safe when is valid: $\forall M \in [M_0\rangle : M(p) \leq 1$ where the $[M_0\rangle$ represents the set of the reachable markings and M_0 represents the initial marking of the Petri net.

Is the given P/T Petri net safe?. Since the Reachability (Coverability) Tree is the finite representation of the set of the reachable markings, with its using we can analyse the safeness. As we can see in the designed tree in Figure 2, it contains a few reachable markings, that exceed the value larger than one, for instance, the initial marking $M_0 = (2,0,2,0,2)$. This is a sufficient condition to declare that this network is **not safe**. □

Boundedness. When each place of the Petri net is bounded, then this net we mark as *bounded*. The place p is *k-bounded* if $\exists k \in N : \forall M \in [M_0\rangle : M(p) \leq k$, where the $[M_0\rangle$ represents the set of the reachable markings and M_0 represents the initial marking of the Petri net. The place is *bounded* if it is *k-bounded* for some $k \in N$. A necessary condition for the k-boundedness of the Petri net is the finality of the initial marking: $M_0 : P \rightarrow \{0, 1, \ldots, k\}$. The each *safe* Petri net is *1-bounded* Petri net.

Is the given P/T Petri net boundedness?. Since the *safeness* and *boundedness* together relate we can this property analysed again using the designed Reachability (Coverability) Tree. The Petri net is said to be *k-bounded* if the number of tokens in each place does not exceed a finite number K for any marking M_n reachable from M_0 (i.e. $M(P) \leq K$). The *P/T Petri net* in Figure 1 is **not bounded**, because the Reachability (Coverability) tree in Figure 2 contains the node with the supremum ω, which represents the possibly infinite number of tokens in given place at relevant marking, respectively the infinite set of the reachable markings. □

Coverability Marking A marking M_1 is said coverable by another marking M_2, when is valid $M_1 \leq M_2$, thus $\forall p \in P : M_1(p) \leq M_2(p)$, where the P represents a finite set of places and M_1 and M_2 are markings of given Petri net.

Is the marking $M_1 = (3,0,1,1,2)$ coverable in given P/T Petri net?. In this task, we can use again Reachability (Coverability) tree, as it was in previous tasks. We want to analyse a tree, whether can be achieved the marking with all number of tokens in all places greater or equal than at places of given marking M_1. After viewing the set of reachable markings we can conclude, that does not exist the marking, which should have the required number of tokens in places $\mathbf{P_3}$ and $\mathbf{P_4}$. When the place $\mathbf{P_4}$ contains token, it means, that the writer currently writes the shared resources and in this case the place $\mathbf{P_3}$ has not contain any token, because all tokens are taken by the writer. We can notice this in the designed tree, where when is token in the place $\mathbf{P_4}$, then never is some tokens in the place $\mathbf{P_3}$ (e.g.: $(2,0,\mathbf{0},\mathbf{1},1)$, $(*,0,\mathbf{0},\mathbf{1},1)$). Therefore, the marking $M_1 = (3,0,1,1,2)$ is **not coverable** by some another marking from the set of reachable markings. □

Liveness. The marking $M \in [M_0\rangle$ is *live*, when $\forall t \in T$ exists some marking $M_1 \in [M_0\rangle$ such, that the transition t is M_1-fireable, where T is the finite set of transitions and $[M_0\rangle$ represents the set of the reachable markings. The Petri net is *live*, when all markings from $[M_0\rangle$ are *live*. On the other hand, the Petri net is *not live*, when its Reachability (Coverability) Tree contains some *terminate* node. The terminate node is represented by the marking M_x for which does not exist some fireable transition $t \in T$.

Can be the given P/T Petri net live?. This property can be easy to analyse with the designed Reachability (Coverability) Tree. As have been said above, when the Petri net is live, it cannot have terminal nodes, thus nodes from which is not fireable any transitions. The designed tree contains only *duplicated* nodes (grey background) and *internal* nodes (colour background), thus no terminal nodes. Since all leaf nodes of the tree are duplicated nodes, the *P/T Petri net* in Figure 1 **can be live**. □

---

b) Assume the given *P/T Petri net* without the transitions `t5`, `t6` and with initial marking $M_O = (3,0,2,0,3)$. Compute *P-Invariants* and with using it determine and verify the given tasks.

To verify the tasks in this subsection we need to obtain the *P-Invariants* of the given *P/T Petri net*. *P-Invariant* is the vector of places $x : P \rightarrow Z$ for which is valid, that $\underline{N}^T \cdot x = \mathbf{0}$, so we obtained it by solving this system of linear equations. $\underline{N}^T$ is the transposed matrix of the Petri net N, also called as an *incidence matrix*. The *incidence matrix* $N : P \times T \rightarrow Z$ is the matrix whose rows correspond to places and whose columns correspond to transitions. Column $t \in T$ denotes how the firing of t affects the marking of the net: $N(t,p) = W(t,p) - W(p,t)$. Notice that a P-Invariant is a vector with one entry for each place, which indicates that the number of tokens in all reachable markings satisfies some linear invariant

Incidence matrix. We constructed the incidence matrix and its transposed matrix of the *P/T Petri net* from Figure 1.

$$\underline{N} = \begin{array}{c} \\ P1 \\ P2 \\ P3 \\ P4 \\ P5 \end{array} \begin{array}{c} \begin{array}{cccc} T1 & T2 & T3 & T4 \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ -1 & -2 & 2 & 1 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \end{pmatrix} \end{array} \qquad \underline{N}^T = \begin{array}{c} \\ T1 \\ T2 \\ T3 \\ T4 \end{array} \begin{array}{c} \begin{array}{ccccc} P1 & P2 & P3 & P4 & P5 \end{array} \\ \begin{pmatrix} -1 & 1 & -1 & 0 & 0 \\ 0 & 0 & -2 & 1 & -1 \\ 0 & 0 & 2 & -1 & 1 \\ 1 & -1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

The natural solution of the equation $\underline{N}^T \cdot x = 0$ such that $x \neq 0$ will be the searched *place invariant (P-Invariant)* of the Petri net.

$$\underline{N}^T \cdot x = 0 \Rightarrow \begin{array}{c} \\ T1 \\ T2 \\ T3 \\ T4 \end{array} \begin{array}{c} \begin{array}{ccccc} P1 & P2 & P3 & P4 & P5 \end{array} \\ \begin{pmatrix} -1 & 1 & -1 & 0 & 0 \\ 0 & 0 & -2 & 1 & -1 \\ 0 & 0 & 2 & -1 & 1 \\ 1 & -1 & 1 & 0 & 0 \end{pmatrix} \end{array} \cdot \begin{array}{c} x \\ \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} \end{array} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \begin{array}{cccc} b & -a & -c & = 0 \\ d & -2c & -e & = 0 \\ 2c & -d & e & = 0 \\ a & -b & +c & = 0 \end{array}$$

P-Invariants. The given *P/T Petri net* from Figure 1 has the *P-Invariants* listed below. The Petri net is cover by *P-Invariants*, when for each place $p \in P$ exists the positive *P-Invariant* x such, that its element is $x(p) > 0$. We can conclude, that given net is covered by positive *P-Invariants* and has the finite initial marking, therefore it is also bounded. Since this Petri net is cover by *P-Invariants*, then exists *P-Invariant* i of this net, whose elements are $i(p) > 0$ for all $p \in P$.

P-Invariants

P1	P2	P3	P4	P5
1	1	0	0	0
0	1	1	2	0
0	0	0	1	1

P-Invariant equations

$$M(P1) + M(P2) = 3$$
$$M(P2) + M(P3) + 2M(P4) = 2$$
$$M(P4) + M(P5) = 3$$

Is the vectors $v_1 = (1,2,1,3,1)$ and $v_2 = (1,2,1,2,1)$ *P-Invariants*?. In this task, we use the property of *P-Invariants*, which says that linear combinations of P-Invariants (i.e. multiplying an invariant by constant or component-wise addition of two or more invariants) will again yield a P-Invariant. If we take a closer look at the first vector v_1, we can see that it was created by a linear combination of all minimal-support P-Invariants, which are linearly independent:

$$L : v_1 = (1,2,1,3,1)$$
$$R : (1,1,0,0,0) + (0,1,1,2,0) + (0,0,0,1,1) = (1,2,1,3,1) \quad \checkmark$$

Based on this property, we can say that this vector v_1 **is the P-Invariant**. For clarity, we also show the use of derived equations from the incidence matrix too, where the components of equations (a,b,c,d,e) are matching with the components of the vector in the same order:

$$\begin{array}{rcll} b - a - c = 0 & \Rightarrow & 2 - 1 - 1 = 0 & \checkmark \\ d - 2 \cdot c - e = 0 & \Rightarrow & 3 - 2 \cdot 1 - 1 = 0 & \checkmark \\ 2 \cdot c - d + e = 0 & \Rightarrow & 2 \cdot 1 - 3 + 1 = 0 & \checkmark \\ a - b + c = 0 & \Rightarrow & 1 - 2 + 1 = 0 & \checkmark \end{array}$$

In contrast, the vector v_2 was probably not composed of any linear combination of the found linearly independent P-Invariants and therefore it cannot be verified only according to this property. The same as in the first vector, we compute the derived equations from the incidence matrix, When the given vector v_2 would be the P-Invariant these equations have to valid:

$$\begin{aligned} b-a-c=0 &\quad\Rightarrow\quad 2-1-1=0 && \checkmark \\ d-2\cdot c-e=0 &\quad\Rightarrow\quad 2-2\cdot 1-1=-1 && \times \\ 2\cdot c-d+e=0 &\quad\Rightarrow\quad 2\cdot 1-2+1=1 && \times \\ a-b+c=0 &\quad\Rightarrow\quad 1-2+1=0 && \checkmark \end{aligned}$$

The vector v_2, therefore, **cannot be P-Invariant** of the given Petri net. □

Strictly Conservation. A Petri net is strictly conservative it the number of tokens is preserved in each marking when transition fires. The Petri net is called *strictly conservative* when is valid the following condition: $\forall M \in [M_0\rangle : \sum_{p\in P} M(p) = \sum_{p\in P} M_0(p)$. The strict conservativeness of the Petri net is a very strong property. Easily it can be seen that for a strictly conservative net have to valid: $\forall t \in T : \sum_{p\in {}^\bullet t} W(p,t) = \sum_{p\in t^\bullet} W(t,p)$. If this condition were not met for some transition t, then it would its execution changed the total number of net tokens.

Is the given P/T Petri net strictly conservative? When the network should be strictly conservative, then have to exist the P-Invariant i, which covers all places with the elements equal to one: $\forall p \in P : i(p) = 1$. As we have shown above, the given Petri net is covered by three linearly independent P-Invariants, therefore this net **cannot be strictly conservative**. For instance, the initial marking $M_0 = (3,0,2,0,3)$ contains in total **8** tokens in places, but after the firing of the transition t_2 the net will have the current marking $M_1 = (3,0,0,1,2)$, what includes only **six** tokens. So we can see that the number of tokens in the net has changed after making the transition and therefore this net **cannot be marked as strictly conservative**. □

Conservation with respect to the weight vector. The lighter condition to the conservativeness of the Petri net, in compare to strictly conservativeness, is connected to the use of the weight vector. The Petri net is conservative if there exists a vector $w : P \to N$, $w = (w_1, w_2, \ldots, w_m)$, where m is the number of places, and $w(p) > 0$ for each $p \in P$ such that the weighted sum of the tokens remains constant: $\forall M \in [M_0\rangle : \sum_{p\in P} w(p)M(p) = \sum_{p\in P} w(p)M(p)$.

Is the given P/T Petri net conservative with respect to the some weight vector? The concept of the *P-Invariants* is largely related to this property of the Petri net, where is verified its conservativeness with respect to the weight vector. Each *P-Invariant* of the Petri net can be considered as a general weight vector, in which are also negative elements allowed. Unlike the concept of strict conservativeness, which accepted the requirement of an unchanging number of tokens across all net, this concept is more general and reflects certain net conditions that should be maintained. The weight vector can cover only some specific places, as same as the *P-Invariants*, and therefore we can define which places have the significant role and for them define the weights, and for remaining places stay its weight equal to zero. For the above reasons, we can conclude that the *P/T Petri net* from Figure 1 **is conservative with respect to the weight vector** a the examples of such vector can be the listed *P-Invariants* (i.e. $(1,1,0,0,0)$, $(0,1,1,2,0)$ and $(0,0,0,1,1)$) or the vector $v_1 = (1,2,1,3,1)$. □

Reachability. In this task, we use the marking of the net with respect to the *P-Invariants*. We have the Petri net with the initial marking M_0, then for each *P-Invariant* i of this net and for each reachable marking $M \in [M_0\rangle$ is valid, that $M \cdot i = M_0 \cdot i$. We already know that P-Invariants keeps the number of tokens at specific places always constant. In this context is significant to observe the number of tokens in the initial marking M_0 and in the current marking M_1, after the multiplying by relevant P-Invariant. The sum of the tokens in these specific places has to be same in both results.

Is the marking $M_2 = (3,0,1,1,2)$ reachable in given P/T Petri net?. About the semantics of the *P-Invariants* we will discuss in the next task, however, to understand the reasoning in this task we introduce the basic information about they. Let us remember the all *P-Invariants* of the given Petri net:

$$i_1 = (1,1,0,0,0) \qquad i_2 = (0,0,0,1,1) \qquad i_3 = (0,1,1,2,0)$$

For the purpose of this solution, we need to say the importance of the individual *P-Invariants*. The *P-Invariant* i_1 cover the places P_1 and P_2, thus the places where the readers waiting, respectively reading and therefore it secures the constant number of readers in the net. In a similar way, the *P-Invariant* i_2 cover the places P_4 and P_5, which have the same meaning for writers, and therefore it maintains the constant number of writers. The last *P-Invariant* cover the places P_2, P_3 and P_4, thus the places where move the tokens which represent the available resources, and therefore this *P-Invariant* secure the unchangeable number of the resources in the net.

We can verify the fact that *P-Invariants* do not change the number of tokens in the specific places, on the defined formula $M_0 \cdot i$, where i represents the relevant *P-Invariants*:

$$\begin{aligned}
M_0 \cdot i_1 &= (\mathbf{3},\mathbf{0},2,0,3)\cdot(\mathbf{1},\mathbf{1},0,0,0) = (\mathbf{3},\mathbf{0},0,0,0) && \checkmark \\
M_0 \cdot i_2 &= (3,0,2,\mathbf{0},\mathbf{3})\cdot(0,0,0,\mathbf{1},\mathbf{1}) = (0,0,0,\mathbf{0},\mathbf{3}) && \checkmark \\
M_0 \cdot i_3 &= (3,\mathbf{0},\mathbf{2},\mathbf{0},3)\cdot(0,\mathbf{1},\mathbf{1},\mathbf{2},0) = (0,\mathbf{0},\mathbf{2},\mathbf{0},0) && \checkmark
\end{aligned}$$

In the same way, we can verify the given marking $M_2 = (3,0,1,1,2)$, for which have to apply the formula $M_0 \cdot i = M_2 \cdot i$ for the all *P-Invariants*, if it should be reachable:

$$\begin{aligned}
M_2 \cdot i_1 &= (\mathbf{3},\mathbf{0},1,1,2)\cdot(\mathbf{1},\mathbf{1},0,0,0) = (\mathbf{3},\mathbf{0},0,0,0) && \checkmark \\
M_2 \cdot i_2 &= (3,0,1,\mathbf{1},\mathbf{2})\cdot(0,0,0,\mathbf{1},\mathbf{1}) = (0,0,0,\mathbf{1},\mathbf{2}) && \checkmark \\
M_2 \cdot i_3 &= (3,\mathbf{0},\mathbf{1},\mathbf{1},2)\cdot(0,\mathbf{1},\mathbf{1},\mathbf{2},0) = (0,\mathbf{0},\mathbf{1},\mathbf{2},0) && ✗
\end{aligned}$$

We can see, that the *P-Invariant* i_3 changed the number of tokens in the places P_2, P_3 and P_4 from **2** to **3** and therefore the given marking M_2 **cannot be reachable** in the *P/T Petri net* from Figure 1. □

Interpret the semantic of the P-Invariants in the P/T Petri net about the Readers and Writers.

P-Invariant $i_1 = (1,1,0,0,0)$**.**

$$\forall M \in [M_0\rangle : M(P_1) + M(P_2) = M_0(P_1) + M_0(P_2) = 3$$

This P-Invariant represents the constant number of the process in the places P_1 and P_2. No processes are lost or added and each process is in one of both places. In these places are located the readers and therefore it represents that the number of readers is constant and each reader wait or read.

P-Invariant $i_2 = (0,0,0,1,1)$**.**

$$\forall M \in [M_0\rangle : M(P_4) + M(P_5) = M_0(P_4) + M_0(P_5) = 3$$

This P-Invariant represents the constant number of the processes in the places P_4 and P_5. No processes are lost or added and each process is in one of both places. In these places are located the writers and therefore it represents that the number of writers is constant. Either all writers wait in the place P_5, or one writer write in the place P_4 and the others wait in the place P_5.

P-Invariant $i_3 = (0,1,1,2,0)$**.**

$$\forall M \in [M_0\rangle : M(P_2) + M(P_3) + 2 \cdot M(P_4) = M_0(P_2) + M_0(P_3) + 2 \cdot M_0(P_4) = 2$$

This P-Invariant represents the constant number of the processes in the places P_2, P_3 and P_4. The place P_4 contains at most the one token, which means, that can exist at most one writer that currently writing. In this case, when the place P_4 contains the token, then the places P_2 and P_3 do not contain any token ($M(P_2) = M(P_3) = 0$). This means when some writers currently writing, then all readers waiting in the place P_1 and none of them does not read in the place P_2 and also no resources are not available in the place P_3. The place P_2 can contain at most 2 tokens, thus at most two readers can read simultaneously from resources. In this case, all writers waiting ($M(P_4) = 0$) and also the place P_3 is empty (no resources available).

c) Assume the given *P/T Petri net* without the transitions `t5`, `t6` and with initial marking $M_O = (3,0,2,0,3)$. Compute *T-invariants* and with using it determine and verify the given tasks.

Similar as in the previous task, to verify this task we need to obtain *T-Invariants* of the given P/T Petri net. A *T-Invariant* is a vector with one entry for each transition $x : T \to Z$, which is a natural solution of the equation $\underline{N} \cdot x = 0$, where the N represents the incidence matrix of the Petri net. Let us have the places $M_0, M_1, \ldots, M_k \in [M_0\rangle$ and the transitions $t_1, t_2, \ldots, t_k \in T$ and the following firing sequence from it: $M_0\,[t_1\rangle\, M_1\,[t_2\rangle \ldots [t_k\rangle\, M_k$. Now, we ca defines the vector x as follows: $x(t) = |\{i : t_i = t \wedge 1 \le i \le k\}|$, and then is valid the equation $M_0 + \underline{N} \cdot u = M_k$. The *T-Invariant* indicates a possible loop in the net, i.e. a sequence of transitions whose net effect is null, i.e. which leads back to the marking it starts in.

Incidence matrix. Let us remember the constructed incidence matrix of the *P/T Petri net* from Figure 1, whose rows correspond to places and whose columns correspond to transitions. Since the *T-Invariants* are the natural solution of the equation $\underline{N} \cdot x = 0$, such that $x \neq 0$, also list this equation:

$$\underline{N} \cdot x = 0 \;\Rightarrow\; \begin{array}{c} \\ P1 \\ P2 \\ P3 \\ P4 \\ P5 \end{array} \begin{array}{c} \begin{array}{cccc} T1 & T2 & T3 & T4 \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ -1 & -2 & 2 & 1 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \end{pmatrix} \end{array} \cdot \begin{array}{c} x \\ \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \end{array} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \begin{array}{cccccc} 0 & 0 & -a & d & = & 0 \\ 0 & 0 & a & -d & = & 0 \\ -a & -2b & 2c & +d & = & 0 \\ 0 & 0 & b & -c & = & 0 \\ 0 & 0 & -b & c & = & 0 \end{array}$$

T-Invariants. The given P/T Petri net from Figure 1 has the *T-Invariants* listed below. The Petri net is cover by *T-Invariants*, when for each transition $t \in T$ exists the positive *T-Invariant* x such, that its element is $x(t) > 0$. Since this Petri net is cover by *T-Invariants*, then exists *T-Invariant* i of this net, whose elements are $i(t) > 0$ for all $t \in T$.

T-Invariants

T1	T2	T3	T4
1	0	0	1
0	1	1	0

Is the vectors $v_1 = (30, 20, 20, 30)$ and $v_2 = (2, 3, 2, 3)$ *T-Invariants*?. Ih this task, we use the property of *T-Invariants*, which says that linear combinations of *T-Invariants* (i.e. multiplying an invariant by constant or component-wise addition of two or more invariants) will again yield the *T-Invariants*. If we take a closer look at the first vector v_1, we can see that it was created by a linear combination of two minimal-support *T-Invariants*, which are linearly independent:

$$L : v_1 = (30, 20, 20, 30)$$
$$R : 30 \cdot (1, 0, 0, 1) + 20 \cdot (0, 1, 1, 0) = (30, 0, 0, 30) + (0, 20, 20, 0) = (30, 20, 20, 30) \qquad \checkmark$$

We can see, that by the multiplying the first *T-Invariant* by **30** and the second *T-Invariant* by **20** and by the subsequent addition of these results we get the vector v_1. Based od this property, we can say that this vector v_1 **is the T-Invariant**. Same as in the previous task, we also show the use of derived equation from the incidence matrix, where the components (a, b, c, d) are matching with the components of the vector in the same order:

$$\begin{array}{rcll} -a + d = 0 & \Rightarrow & -30 + 30 = 0 & \checkmark \\ a - d = 0 & \Rightarrow & 30 - 30 = 0 & \checkmark \\ -a - 2 \cdot b + 2 \cdot c + d = 0 & \Rightarrow & -30 - 2 \cdot 20 + 2 \cdot 20 + 30 = 0 & \checkmark \\ b - c = 0 & \Rightarrow & 20 - 20 = 0 & \checkmark \\ -b + c = 0 & \Rightarrow & -20 + 20 = 0 & \checkmark \end{array}$$

In contrast, the vector v_2 was probably not composed of any linear combination of the found linearly independent *T-Invariants* and therefore it cannot be verified only according to this property. The same as in the first vector, we compute the derived equations from the incidence matrix. When the given vector $v_2 = (2, 3, 2, 3)$ would be the *T-Invariant* these equations have to valid:

$$
\begin{aligned}
-a + d = 0 \quad &\Rightarrow \quad -2+3 = 1 & ✗\\
a - d = 0 \quad &\Rightarrow \quad 2-3 = -1 & ✗\\
-a - 2\cdot b + 2\cdot c + d = 0 \quad &\Rightarrow \quad -2-2\cdot 3+2\cdot 2+3 = -1 & ✗\\
b - c = 0 \quad &\Rightarrow \quad 3-2 = 1 & ✗\\
-b + c = 0 \quad &\Rightarrow \quad -3+2 = -1 & ✗
\end{aligned}
$$

The vector v_2, therefore, **cannot be T-Invariant** of the given Petri net. □

What can be determined from the calculated T-invariants about the liveness of the Petri net and why?

In this task we will discuss the role of the *T-Invariants* in the live bounded Petri nets. When the Petri net would be *live* and *bounded*, then exists the marking $M \in [M_0\rangle$ and the firing sequence $M\,[t_1^{`}\rangle\,M_1\,[t_2^{`}\rangle \ldots [t_k^{`}\rangle\,M_K$, in which each transition $t \in T$ is fired at least once time and where $M_k = M$. Let us define the vector $u : T \to N$ as follows: $u(t) = |\{i : t_i^{`} = t \ \wedge\ 1 \leq i \leq k\}|$. As we already know is valid that $M + \underline{N} \cdot u = M_K$, and since the places equals $M = M_k$, then is valid $\underline{N} \cdot u = 0$. The vector u is the *T-Invariant* and moreover it cover the Petri net since is valid that $\forall t \in T : u(t) > 0$.

We can conclude, that each *live* and *bounded* Petri net is cover by the *T-Invariants*. This is only a necessary condition for liveliness the Petri net, thus the Petri net, which is not covered by the *T-Invariants*, it cannot be *live* and *bounded* simultaneously. The *P/T Petri net* from Figure 1 is covered by positive *T-Invariants*, therefore it **might be live**. □

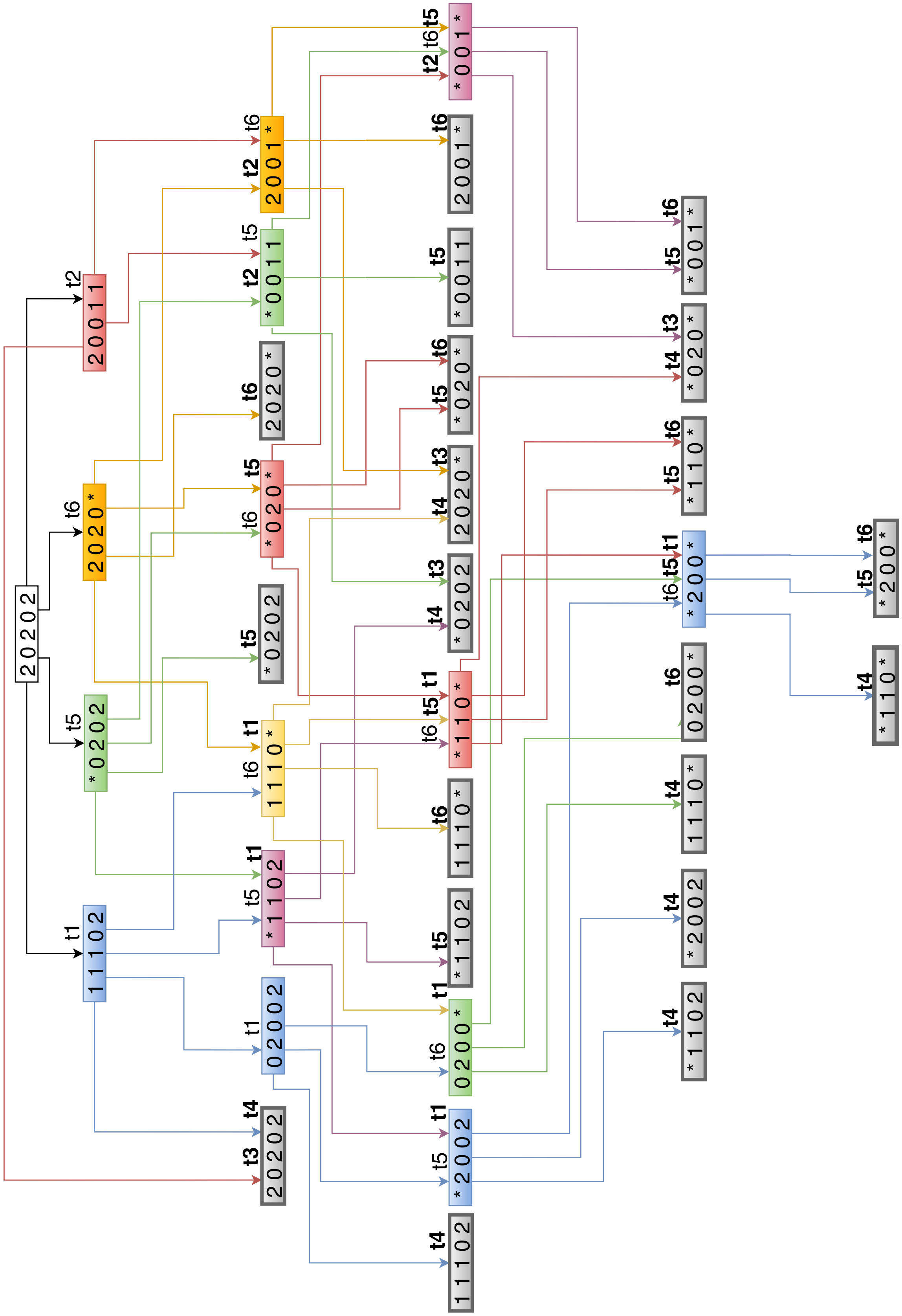

Figure 2. The Reachability (Coverability) Tree of the *P/T Petri net* from Figure 1.

2. TASK

Assignment: Model the *P/T* Petri net for a Dijkstra algorithm for mutual exclusion. The algorithm listed below assumes the non-limited number of parallel processes, which with a unique identification, and describes the algorithm for the process with identifier *i*. The array of boolean values `flag` and the boolean variable `p` are shared between all processes, and they are initialised to the 0. Model the system with just two processes with indices 0 and 1 (modelling of unique indices for non-limited number of the processes do not within the power of *P/T* Petri nets).

Algorithm 1: Process with index *i*

```
while true do
    flag[i] := true;
    if p ≠ i then
        wait until not flag[p];
        p := i;
    if ∃j : j ≠ i ∧ flag[j] then continue ;
    critical_section;
    flag[i] := false;
```

2.1 Task

Model the system in the tool `Netlab`. Use the modelling techniques, where the places of the *P/T* Petri net represents the lines of the algorithm and the values of the variables. Try to make the model as read and clear as possible, use the text marking of the places and transitions. Perform the available analysis in the `Netlab` and interpret the obtained results. Based on the obtained results determine the answers on the following questions:

(a) Does guarantee the protocol *mutually exclusion* (i.e., processes cannot be simultaneously in the *critical section*)?

(b) Does guarantee the protocol impossibility of deadlock?

2.2 Task

What happens if we allow a process code with indexes 0 or 1 to execute by unlimited processes at the same time? Try it in `Netlab`.

Model description. The designed model is saved in the file `uloha1.net`, which is available on the root level of this archive. The set of the places $\{P_1, P_2, \dots, P_8\}$ represents the code of the first process with index $i = 0$ in the same order as they are in Algorithm 1. The set of the places $\{P_{15}, P_{16}, \dots, P_{22}\}$ represents the code of the second process with index $i = 1$ in the same order as they are in Algorithm 1. The pair of places (P_9, P_{10}) represents the value of the first item in the `flag` array (`flag[0]`), the next pair (P_{13}, P_{14}) represents the value of the second item of this array (`flag[1]`). The remaining two places (P_{11}, P_{12}) represents the value of the boolean variable *p*, which can contain the value 0 or 1. The transitions $\{T_1, T_2, \dots, T_{11}\}$ belong to process with index $i = 0$ and the remaining transitions $\{T_{12}, T_{13}, \dots, T_{22}\}$ belong to process with index $i = 1$. The individual places and transitions are named according to its feature from the given Algorithm 1.

Model analysis.

We performed the available analysis of the designed *P/T Petri net* within the tool `Netlab` and now we gradually introduce and determine the obtained results.

Boundedness. As we defined in few sections above, the *P/T Petri net* is *bounded*, when it is cover by *P-Invariants* and has the finite initial marking M_0. Also, we known, that the *P/T Petri net* is cover by the *P-Invariants* when for each its place $p \in P$ exists the positive *P-Invariant*. According to the performed analysis, we can conclude, that there exist **5** positive *P-Invariants* which cover all places of the designed *P/T Petri net*. Therefore, the sufficient condition for boundedness is satisfied, and since the net has also the finite initial marking M_0, we can conclude, that designed net **is bounded**. We also use to analyse this property the *Reachability (Coverability) tree*, whose nodes cannot contain ω as an element of the markings, when the given Petri net should be *boundedness*.

Safeness. The *P/T Petri net* is *safe*, when each place is *safe*. The place $p \in P$ is safe when for each marking $M \in [M_0\rangle$ is valid $M(p) \leq 1$, where M_0 represents again the initial marking of the Petri net. According to the obtained *Coverability* and *Reachability* graph, respectively the set of the all reachable markings ($[M_0\rangle$), we can see, that the number of tokens in each place is never larger than one and therefore they can be represented as a *binary vector*. This property could also be analysed on the basis that the capacity of any of the places is not greater than one. Thus, the designed *P/T Petri net* is also **safe**. Note that this property is only a special case of *k-boundedness* when $k = 1$.

Conservation. In the *conservative P/T Petri net* remains the constant number of tokens in the whole net after the firing of each transition. When each marking from the set of the *reachable markings*, obtained from the *Reachability (Coverability) tree*, include the equal number of tokens and does not contain ω, then the net can be marked as *conservative*. In the case of *strict conservativeness*, we can analyse the *P-Invariants* to verify it. If the net should be *strictly conservative*, then have to exist the binary *P-Invariant i*, which covers all places of the given net: $\forall p \in P : i(p) = 1$. Based on the obtained *P-Invariants* from the *Netlab*, we can conclude, that this *P/T Petri net* **is strictly conservative** and during the whole run it contains the constant number of tokens. If we think about it, then the tokens are at the beginning of the run in the starting places of the processes (P_1, P_{15}) and also at the initial places of the variables (P_{10}, P_{12}, P_{13}). Both processes have to be in one place during the whole run and the values of the variables have to be defined therefore they will be in one place too. This makes it clear that the number of tokens remains constant, always **5 tokens**.

Liveness. The *P/T* Petri net with initial marking M_0 is *live* if, no matter what marking has been reached from initial marking, it is possible to ultimately fire any transition by progressing through some further firing sequence. A live Petri net guarantees *deadlock-free* operation, no matter what firing sequence is chosen. Based on the *Reachability (Coverability) tree* is possible to verify this property check the structure of tree, whether it includes some terminate node. When the constructed tree includes some such nodes, then we can say, that given net is not *live*. As we already defined, the next way to analyse the *liveness* of the *P/T Petri net* is using the *T-Invariants*. Each *live* and *bounded* Petri net is cover by *T-Invariants*, this reflects the necessary condition for liveness of the net. Now, we say, that analysed *P/T Petri Net* **is not live**, but we will analyse this property in more detail in the following sections.

Partial deadlocks. Firstly, we will analyse the constructed *Reachability tree* and its impact to analysed features. When we look at its condensed version, we can see, that this tree includes two terminal nodes (K_3, K_7), what violates the condition for *liveness* of the Petri net. Both these nodes cause the *partial deadlocks* in the Petri net, when the one process cycles to infinity within four places and the second one is trapped in one place without the possibility of transition.

The first *partial deadlock* occurs after the firing of the following transition sequence: $(T_1, T_3, T_{12}, T_{13}, T_{15})$, where the current marking of the net will be the following $M_{K_3} = (P_5, P_9, P_{12}, P_{14}, P_{18})$ [1]. In this situation, the second process with index $i = 1$ has not any possible firing transition from the place P_{18} and the first process (index $i = 0$) have to repeat the following sequence of transition (T_5, T_8, T_1, T_2) in this order between the places (P_3, P_6, P_1, P_2), what represents the repeating of the algorithm lines with numbers $(3, 6, 1, 2)$. While the second process waiting until the first process drops own flag to 0, the first process cannot enter to the critical section because of a raised flag of the second process.

The second *partial deadlock* is caused by the same occurrences only in the reverse order of the processes. After the firing of transition sequence $(T_1, T_{12}, T_{15}, T_{17}, T_{18}, T_4)$ will be the current marking of the net following $M_{K_7} = (P_4, P_9, P_{11}, P_{14}, P_{20})$. The second process will repeat the sequence of transitions $(T_{19}, T_{12}, T_{14}, T_{16})$ between the places $(P_{20}, P_{15}, P_{16}, P_{17})$ and the first process will remain blocked in the place P_4 without the option of transition. We emphasise that in both cases the transition sequences were only exemplary, and it would be possible to get into these situations with another transition sequence.

Does guarantee the protocol impossibility of deadlock?

As we have introduced, the *liveness* of the *P/T Petri Net* is close related to the *deadlock*, because if the net is not *live*, that exists some marking which causes *deadlock*. Although, there exists a positive *T-Invariants*, which makes, that the necessary condition for liveness is satisfied and the net may be *live*, in this case it will not be so. Since there are exists two *partial deadlocks* this net **cannot be live**. In the *P/T Petri net* from Figure 1 there **will never be total deadlock** of both processes, but there can arise the situation with **partial deadlocks** of one process. Thus, we can conclude, that the protocol **guarantees** the impossibility of *total deadlock*, but **does not guarantee** the impossibility of *partial deadlock*.

Does guarantee the protocol mutually exclusion (i.e., processes cannot be simultaneously in the critical section)?

To verify this property of the designed *P/T Petri Net* we add next place to the model, which serves for this purpose. The place P_{23} (`mutual_exclusion_test`) contains the token when the critical section is free. Our goal is to verify the

[1] For better orientation, we only list the places where the token is currently located.

mutual exclusion of the critical section, thus, the at most one process is present here. By the addition of this place to the net arise the new *P-Invariant*, which cover this place and two places, where are processes present in the critical section: $M(P_7)+M(P_{21})+M(P_{23}) = 1$. Based on this *P-Invariant* we can conclude, that the protocol **guarantees the mutual exclusion**. This feature could also be verified using the *Reachability (Coverability) Tree*, respectively with its nodes. When we want to verify the mutual exclusion of two places, which represent the critical section of two processes, then we can check this tree whether on specific places are in some marking present two tokens simultaneously. In the obtained *Reachability (Coverability) Tree* from *Netlab* we can see, that in the relevant places ($\mathbf{P_7}$ and $\mathbf{P_{21}}$) no two tokens are present at the same time. So we have also shown that this protocol **guarantees mutual exclusion**.

What happens if we allow a process code with indexes 0 or 1 to execute by unlimited processes at the same time? Try it in `Netlab`.

When we allow a code of both processes to execute by unlimited number of processes at the same time, then the following changes occur. Clearly, since we generate the unlimited number of processes, such modified *P/T Petri net* **will not be bounded and safe**. With this net the protocol loss the guarantees of the *mutual exclusion* from the following reason. Consider a situation where are several processes in the critical section of process with index **0** and processes with an index **1** all waiting for the condition where the contrary flag will be down. Subsequently, when the one process with index **0** will leave the critical section, at the same time he drops his flag to **0** and by this, it will allow the entry of the processes with index **1** to the critical section. Obviously, that in such moment will be in the critical section processes with indices **0** and **1** and therefore the **mutual exclusion will be violated**. The protocol but still **guarantees** the impossibility of *total deadlock*, however, it **does not guarantee** the impossibility of *partial deadlock*, which was described above and which may also occur at this modification.

3.TASK

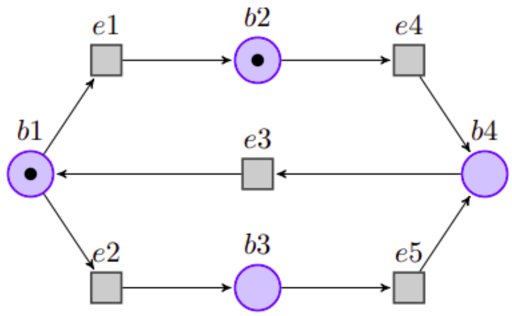

Figure 3. C/E System.

Assignment: For a *C/E System* on Figure 3 perform the given tasks.

2. Complement the system

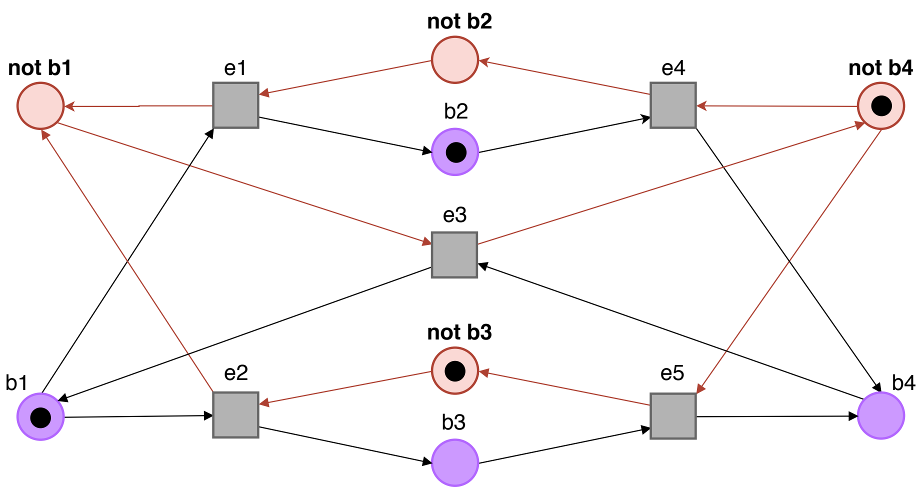

Figure 4. The complement of the C/E System from Figure 3. The *red* colour marks the new elements of the C/E system.

3. Create the *Case Graph*

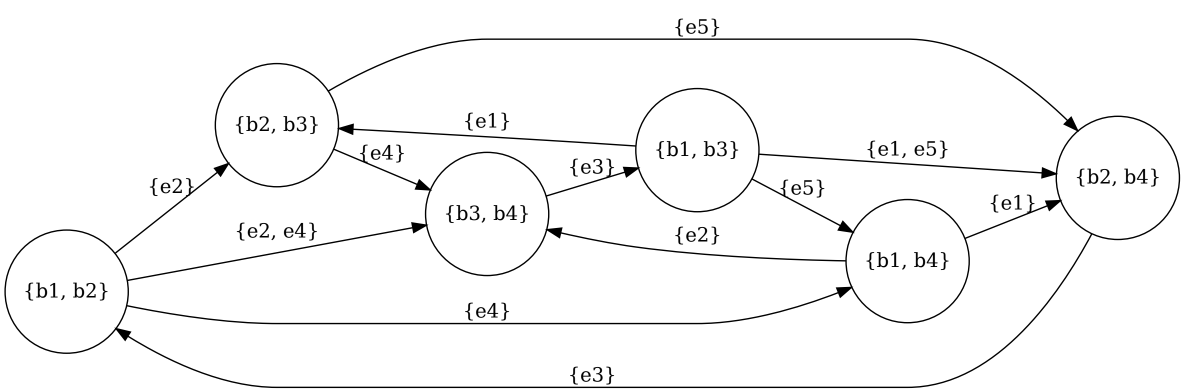

Figure 5. The Case Graph of the C/E System from Figure 3.

1. Decide which of the following formulas is valid and draw it using facts.:

(a) $(\neg b1 \rightarrow (\neg b4 \rightarrow (\neg b2 \rightarrow \neg b3)))$

(b) $(b1 \wedge b2) \rightarrow (b2 \vee b4))$

Firstly, we need to verify the validity of the given formulae, thus if they are valid for each *case* of the *C/E System*, because then they are valid in the whole system. Such formulas are called tautologies and it describe the invariant properties of the *C/E System*. When the formula is valid in the whole *C/E System* then it can be represented as a *fact*, which is not fireable in any case of the system and therefore they does not affect the system. We verify the validity of the given formulae in all *cases* of the analysed *C/E System*, which are available from the constructed *Case Graph* (see Figure 6):

	b1	b2	b3	b4
{b1, b2}	1	1	0	0
{b2, b3}	0	1	1	0
{b3, b4}	0	0	1	1
{b1, b3}	1	0	1	0
{b1, b4}	1	0	0	1
{b2, b4}	0	1	0	1

$(\mathbf{b1} \wedge \mathbf{b2})$	$\rightarrow$	$(\mathbf{b2} \vee \mathbf{b4})$
1	1	1
0	1	1
0	1	1
0	1	0
0	1	0
0	1	1

$(\neg\mathbf{b1}$	$(\neg\mathbf{b4} \rightarrow$	$(\neg\mathbf{b2} \rightarrow \neg\mathbf{b3})))$
1	1	1
1	1	1
1	1	0
1	0	0
1	1	1
1	1	1

From the table, we can conclude, that both formulae **are valid** in all cases of the analysed *C/E Systems* and therefore we can model they as *facts*. To model the first formula ((a)) we rewrite it on the base of the following equivalence rule of implication: $a \rightarrow b \;\Leftrightarrow\; \neg a \vee b$. When we apply this rule to the first formula, then we will get the following form of it:

$$(\neg b1 \rightarrow (\neg b4 \rightarrow (\neg b2 \rightarrow \neg b3))) \Leftrightarrow (\neg b1 \rightarrow (\neg b4 \rightarrow (b2 \vee \neg b3))) \Leftrightarrow (\neg b1 \rightarrow (b4 \vee b2 \vee \neg b3))$$

For the *fact* t is its formula $a(t)$ defined with using of its *pre-condition* and *post-condition* sets, thus when ${}^\bullet t = \{b_1, b_2, \dots, b_n\}$ and $t^\bullet = \{b_1^{'}, b_2^{'}, \dots, b_m^{'}\}$, then $a(t) = (b_1 \wedge b_2 \wedge \dots \wedge b_n) \rightarrow (b_1^{'} \vee b_2^{'} \vee \dots \vee b_m^{'})$. Now we can see that both formulas are in this desired form and the modelling of the facts will be relatively straightforward. Since the first formula contains the *events*, that are not part of the original system ($(\neg b1, \neg b3)$), we use the constructed complement of this system (see Figure 4), which contains these events. We can complement the conditions b_1 and b_3, because the condition of equality of the two *C/E Systems* will not be violated. Equivalent *C/E Systems* always have the same number of events, cases and steps and they can vary in the cardinality of sets of conditions. Since we add only complement conditions, all these sets will remain from the original *C/E System*.

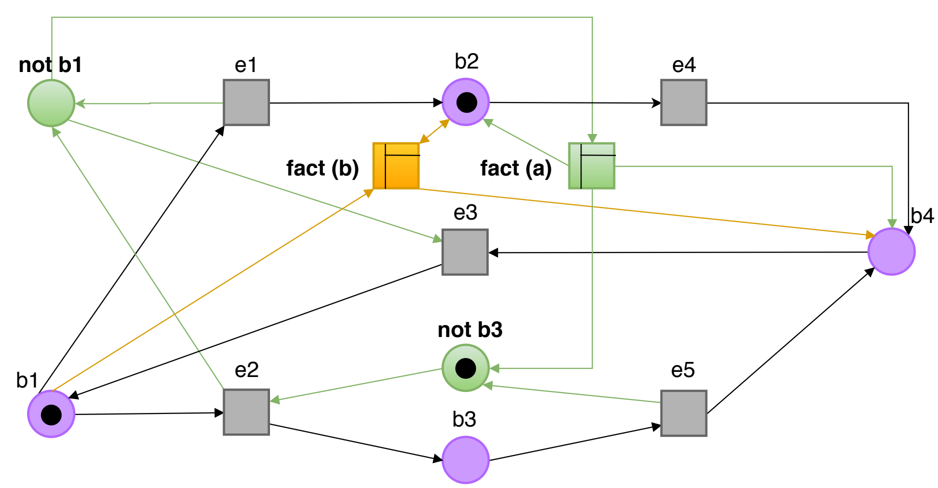

Figure 6. The *C/E System* including the facts representing the given formulae. The orange elements represent the newly added edges due to the fact modelling the second formula (b). The green elements represent the fact modelling the first formula (a), because of which was added two new conditions and also edges connected to it.

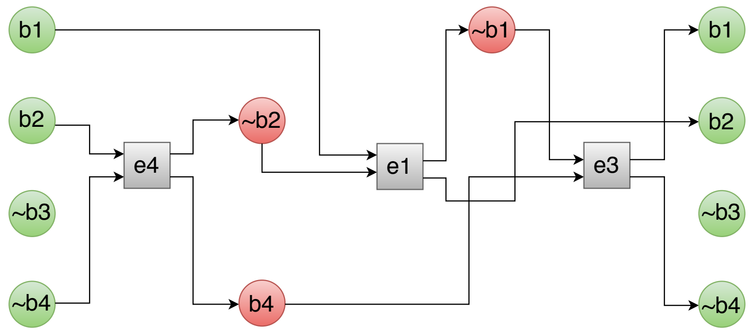

Figure 7. The Occurrence net of the Process 1, which represents the following sequence:
$\{b1, b2, \neg b3, \neg b4\}\,[e4\rangle\,\{b1, \neg b2, \neg b3, b4\}\,[e1\rangle\,\{\neg b1, b2, \neg b3, b4\}\,[e3\rangle\,\{b1, b2, \neg b3, \neg b4\}$.

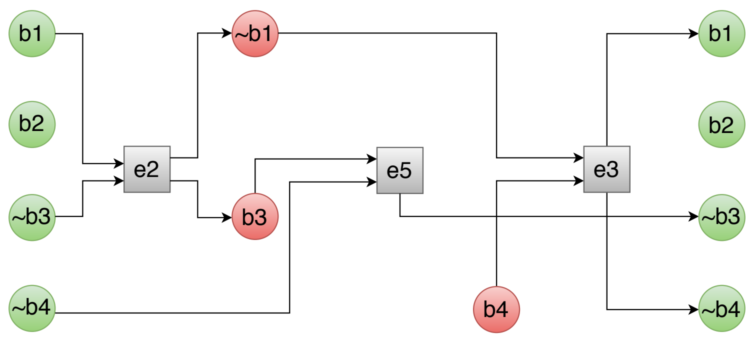

Figure 8. The Occurrence Net of the Process 2, which represents the following sequence:
$\{b1, b2, \neg b3, \neg b4\}\,[e2\rangle\,\{\neg b1, b2, b3, \neg b4\}\,[e5\rangle\,\{\neg b1, b2, \neg b3, b4\}\,[e3\rangle\,\{b1, b2, \neg b3, \neg b4\}$.

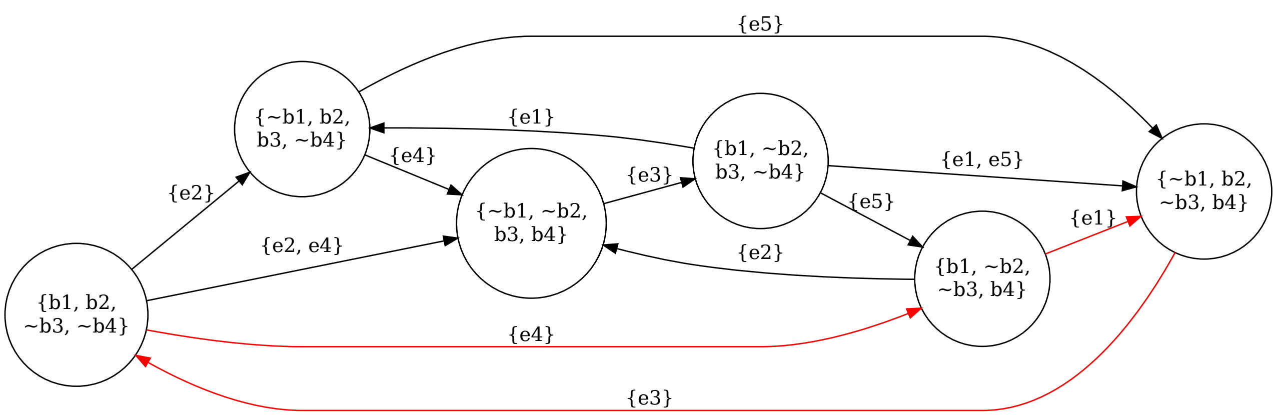

Figure 9. The Case Graph of the C/E System from Figure 4 with the red colour path of the **Process 1** from Figure 7.

4. For complemented *C/E System* draw the shortest process, where is the one case present twice (has two different *S-cuts*, which shows to the same case).

5. Which paths in the Case Graph correspond to this process?

Addendum to the solution of the tasks 4 and 5. As we can see above, we presented two processes (Figure 7 and Figure 8) according to the given requirements. Both presented processes satisfy these requirements, thus they are shortest processes with the repeated case. They have been obtained by analysis the *Case graph* from Figure 6, respectively analysis the whole *C/E System* from Figure 4. Process **1** starts in the initial case $\mathbf{C_0} = \{\mathbf{b1}, \mathbf{b2}, \neg\mathbf{b3}, \neg\mathbf{b4}\}$ and by executing the following sequence of the events $\mathbf{e4} \rightarrow \mathbf{e1} \rightarrow \mathbf{e3}$ it gets back to this initial case $\mathbf{C_0}$. Process **2** also starts in initial case $\mathbf{C_0}$ and the following sequence of the events $\mathbf{e2} \rightarrow \mathbf{e5} \rightarrow \mathbf{e3}$ also get it back to the case $\mathbf{C_0}$. We note, that the whole sequences of both processes are listed in the individual description of Figures 7 and 8. Thus, both processes perform three events and contain the initial case $\mathbf{C_0}$ twice. No other process that was shorter and would contain the same case is not available on this *C/E System*.

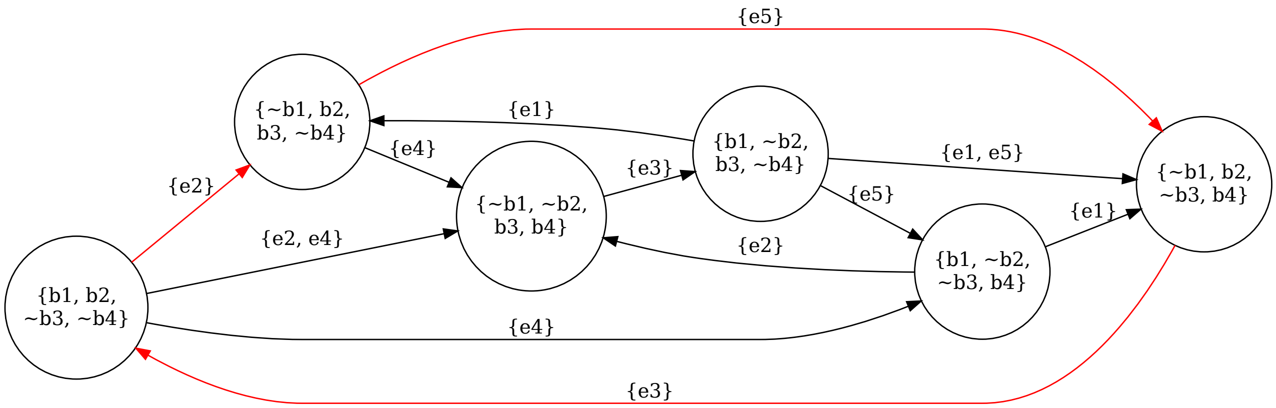

Figure 10. The Case Graph of the C/E System from Figure 4 with the red colour path of the **Process 2** from Figure 8.

The green coloured conditions in the graph of processes represent its values in the initial case $\mathbf{C_0}$ and the red coloured conditions represent the different value of the individual condition in comparison to its value in the initial case $\mathbf{C_0}$. The *Case Graphs* on Figures 9 and 10 represents the individual paths of the presented processes, highlighted in red colour.