

TRANSFORMING NETS

Tadeusz KRUPA

Faculty of Management
Warsaw University of Technology, 02-524 Warszawa, Poland
e-mail: t.krupa@wz.pw.edu.pl

Abstract: T-nets are constructed on the basis of experience gathered during experiments performed on the flow models of production systems [1, 4 and 6-12] and experience from colored Petri nets simulations. Transforming net is functioning in an asynchronous manner due to the initiation of transformers in specific periods of time. Possibilities to time-flow tasks are imminent features of T-nets – resources are gathered in the accumulators; resource transformations are described by the transformers. Task decomposition and task result synthesis is performed in a continuous manner. T-net plays the role of virtual model of real task realization processes. The existence of such model allows to monitor and assess the progress of realized tasks as well as calculations and simulation of task realization strategy.

Key words: accumulator, alternative and parallel dispersing and gathering transformers, colored and transforming Petri net, correctness and functioning, multitask transforming net, task decomposition, task realization strategy, transforming net (t-net), transforming network design.

1 Introduction

Transforming net refers directly to the concept of organizational scheme task [4-6 and 13]. Task's operational scheme is represented by a network of partial tasks connected with parallel and alternate connections (gathering and dispersing). In case of transformational nets one is dealing with the change of resource level and tasks related to it, connected with network operators. Tasks and resources are represented by event identifications (markers) in the transforming network (T-net), whereas task realization process is supervised through markers flowing through network accumulators A and markers performed by the T transformers.

T-net is constructed on the basis of experience gathered during experiments performed on the flow models

of production systems [1, 4 and 6-12] and experience from colored Petri nets simulations. It seems to eliminate ambiguity met in event modeling performed with marker flow through positions and transitions in Petri nets. Possibility to apply many practical constraints on its structure is one of the essential T-net advantages.

In general, the scheme illustrating process flow in selected object can be similar to the one presented in the Figure 1a. Tasks (processes) flow to the object from the left side (marked as: a, b, c, d, ...); task (process) realization results leave the object from the right side (marked as: w^a , w^b , w^c , w^d , ...); process tasks are planned from the top (marked as: p^a , p^b , p^c , p^d , ...) and the planning results are flowing downwards (marked as: r^a , r^b , r^c , r^d , ...).

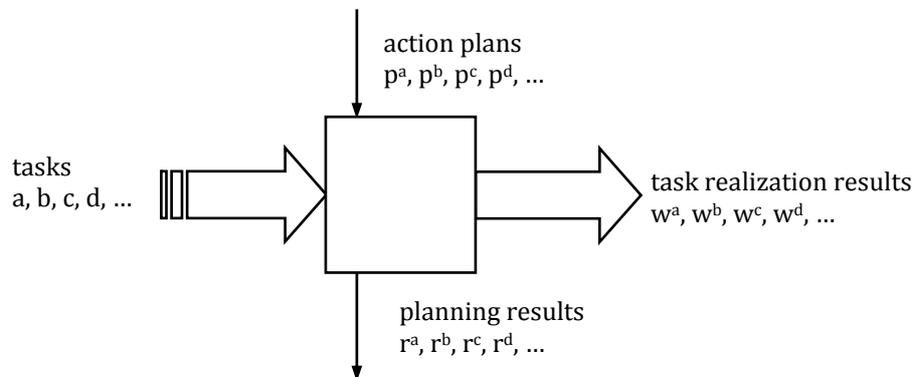


Figure 1a. Task processing scheme (source: self study)

The two main factors of a T-net are its accumulators A_i (see Figure 1b) and alternative $T_j^{V<}$, $T_j^{V>}$ (see Figure 1c) and parallel $T_k^{\&<}$, $T_k^{\&>}$ (see Figure 1d) transformers.

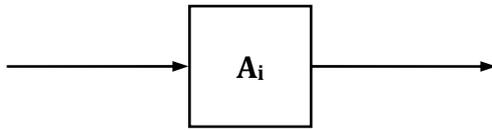


Figure 1b. T-net accumulator (source: self study)

Gathering and forwarding the incoming markers is the main task of the A_i accumulator. The accumulator has only one input and one output, which receives and forwards markers to transformers connected to it. Accumulator's capacity can be limited to a given number of markers.

Alternative transformers (see Figure 1c) can be divided to:

- dispersing alternative transformers $T_j^{V<}$,
- gathering alternative transformers $T_j^{V>}$.

Parallel transformers (see Figure 1d) can be divided to:

- dispersing parallel transformers $T_j^{\&<}$,
- gathering parallel transformers $T_j^{\&>}$.

Modeling of transforming processes is presented in the following structure:

- elements of transforming net structure (see point 2),
- transforming net functioning (see point 3),
- transforming net and color Petri nets (see point 4),
- multitask transforming net (see point 5),
- transforming net modeling (see point 6),
- design of transforming nets (see point 7).

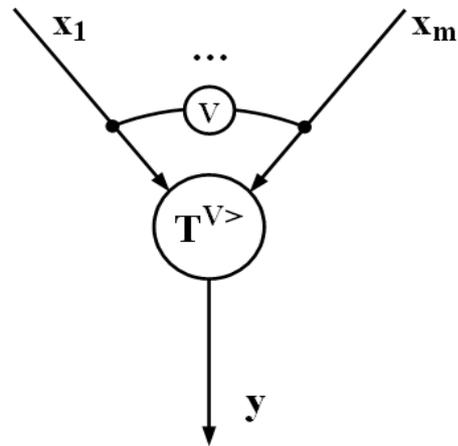
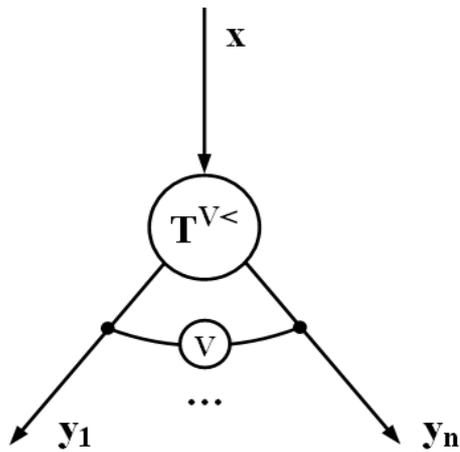


Figure 1c. T-net alternative transformers (source: self study)

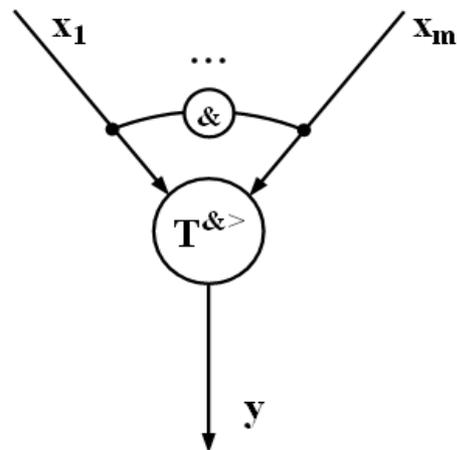
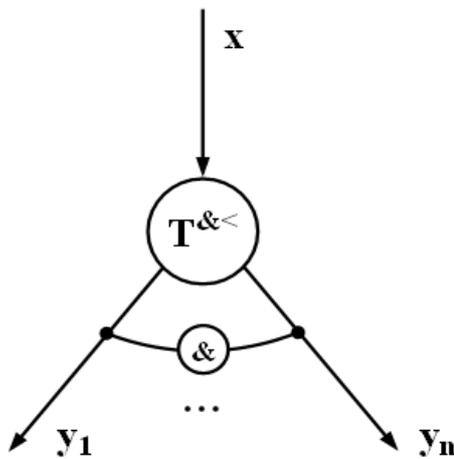


Figure 1d. T-net parallel transformers (source: self study)

2 Elements of transforming network structure

Transforming net (T-net) is constructed from five types of elements:

- dispersing alternative $T^{V<}$ and gathering $T^{V>}$ transformers as well as parallel dispersing $T^{\&<}$ and gathering $T^{\&>}$ transformers, which perform operations on task identifications (markers) that are present in transformer supplying accumulators; event markers are relevant to tasks and resources connected with them (see Fig. 1a),
- accumulators A_x , which take and forward event markers x to T^V and $T^{\&}$ transformers connected with them (see Figure 2a and Figure 2b); certain accumulator of x type can store different event markers (connected with certain events),
- arcs that connect transformers and accumulators,
- event markers,
- functions that are responsible for marker processing in the transformers and for rules that order markers in the accumulators.

Canonical (basic) T-net schematics are presented in the Figure 2a and Figure 2b. Symbols $\&$ and V represent relations between the accumulators and transformers. Circled $\&$ and V symbols are omitted in graphic T-net schematics because of the specific character of marker processing by, so called, transformer operating functions. The V symbol stands for the relations between transformer incoming and outgoing arcs and determine the necessity for an alternative (single – see Figure 2e and Figure 2f) introduction (or leading out) of a marker from

a single input or output of the transformer. The $\&$ symbol stands for the relations between transformer incoming and outgoing arcs and determine the necessity for parallel (simultaneous) introduction (or leading out) of a marker from all inputs or outputs of the transformer (single – please see Figure 2g and Figure 2h).

Possible sub-schematics created on the basis of canonical T-net schematics are presented in the Figure 2c.

Important structural features of the T-nets are as following:

- transformers cannot be directly connected,
- accumulators cannot be directly connected,
- accumulator can have only one input and one output,
- transformer can only have one input and many outputs or single output and many inputs,
- properly constructed net should have at least one starting accumulator without any input (see Figure 2d),
- properly constructed net should have at least one finishing accumulator without any output (see Figure 2d).

Listed features were elaborated on the basis of: task operational schematics function analysis [1, 5 and 13], Petri net model [11] and with the rules of characterization theory taken into consideration [2, 3].

Event markers can be described with letters a, b, c, ... preceded with a number that indicates how many times did the marker appear. Partial marker appearance is not allowed.

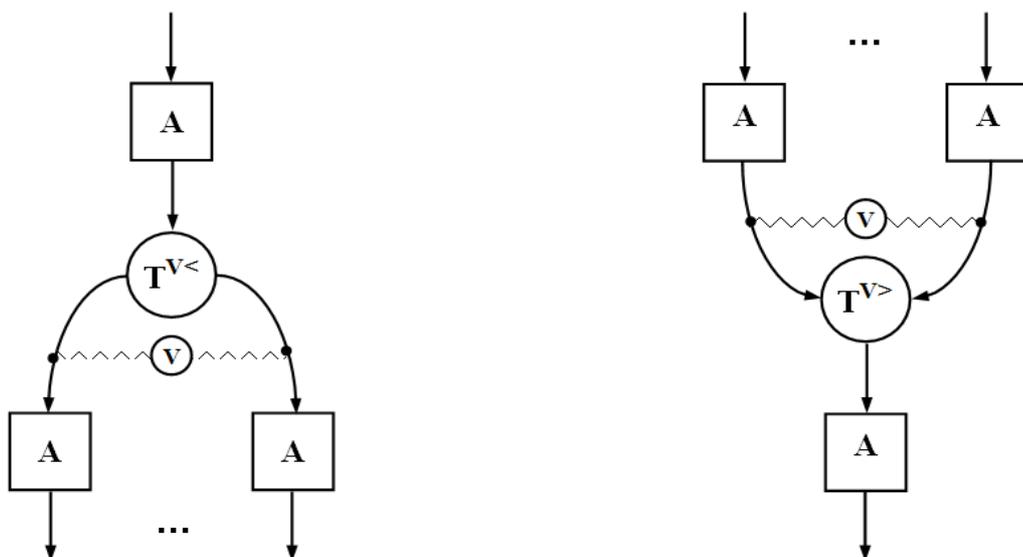


Figure 2a. Accumulators and transformers $T^{V<}$ and $T^{V>}$

(source: self study)

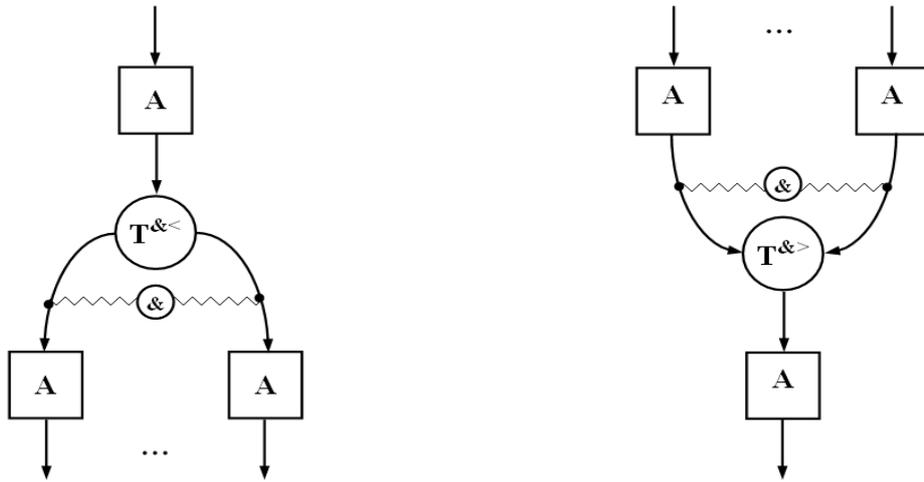


Figure 2b. Accumulators and transformers $T^{\&<}$ and $T^{\&>}$
(source: self study)

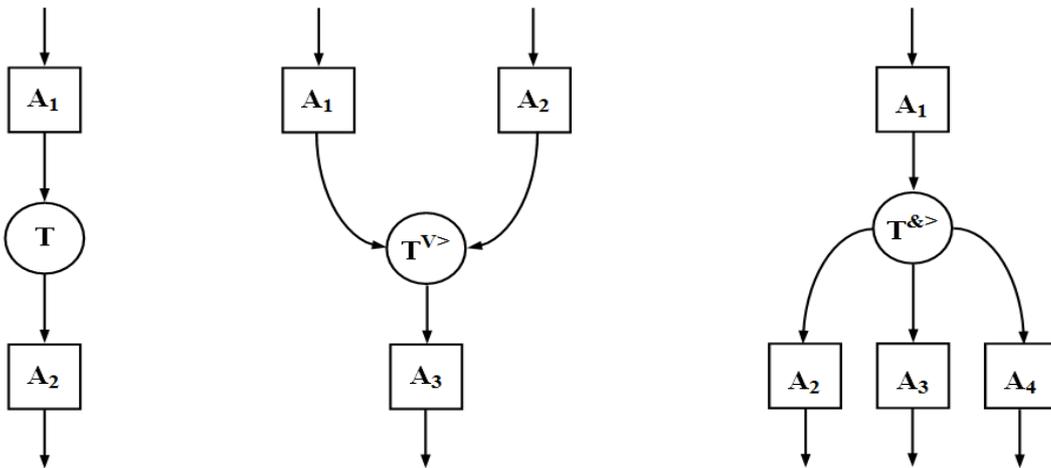


Figure 2c. Examples of T-nets
(source: self study)

In general, functions responsible for marker in transformers processing, so called internal transforming functions, have the following form:

- internal transforming function $T^{\&>int}$
 $T^{\&>int} = \{ \&_{in} x_i \rightarrow y_i \}$
- internal transforming function $T^{V>int}$
 $T^{V>int} = \{ V_{in} x_j \rightarrow y_j \}$
- internal transforming function $T^{\&<int}$
 $T^{\&<int} = \{ x_i \rightarrow \&_{out} y_i \}$
- internal transforming function $T^{V<int}$
 $T^{V<int} = \{ x_j \rightarrow V_{out} y_j \}$

where:

V_{in} - input accumulator alternative selection operation symbol,

V_{out} - output accumulator alternative selection operation symbol,

$\&_{in}$ - simultaneous (conjunctive) input accumulator selection symbol,

$\&_{out}$ - simultaneous (conjunctive) output accumulator selection symbol,

x, y - event markers,

i, j - arc indexes connected conjunctively and alternatively with the transformer.

Internal transforming functions decide about the T-net functioning, which describes the process of transformation of incoming and outgoing event markers. Due to the process of gathering the markers in accumulators,

process course and transformation is conditioned by two factors:

- availability and order (priorities) of markers in the accumulators,
- initiation of marker transformation (priorities) according to certain plan or strategy.

Examples of rules used to order the event markers in the accumulators:

- LIFO rule (markers that were last in the accumulator are served first),
- FIFO rule (markers that were first in the accumulator are served first),
- priority rule (markers with highest priorities, resulting from specific task realization plan, are served first).

The main task of a dispersing alternative transformer $T_j^{V<}$ is to take one marker from the connected accumulator (input marker) and transform it to a single outgoing marker and simultaneously, after certain τ_j time delay, transfer it to a connected accumulator through one of alternative outputs of the $T_j^{V<}$ transformer.

The main task of a gathering alternative transformer $T_j^{V>}$ is to take one input marker from one of connected accumulators and transform it to a single outgoing marker and simultaneously, after certain τ_j time delay, transfer it to a connected accumulator through one of alternative outputs of the $T_j^{V>}$ transformer. Process flow is presented in the Figure 2f.

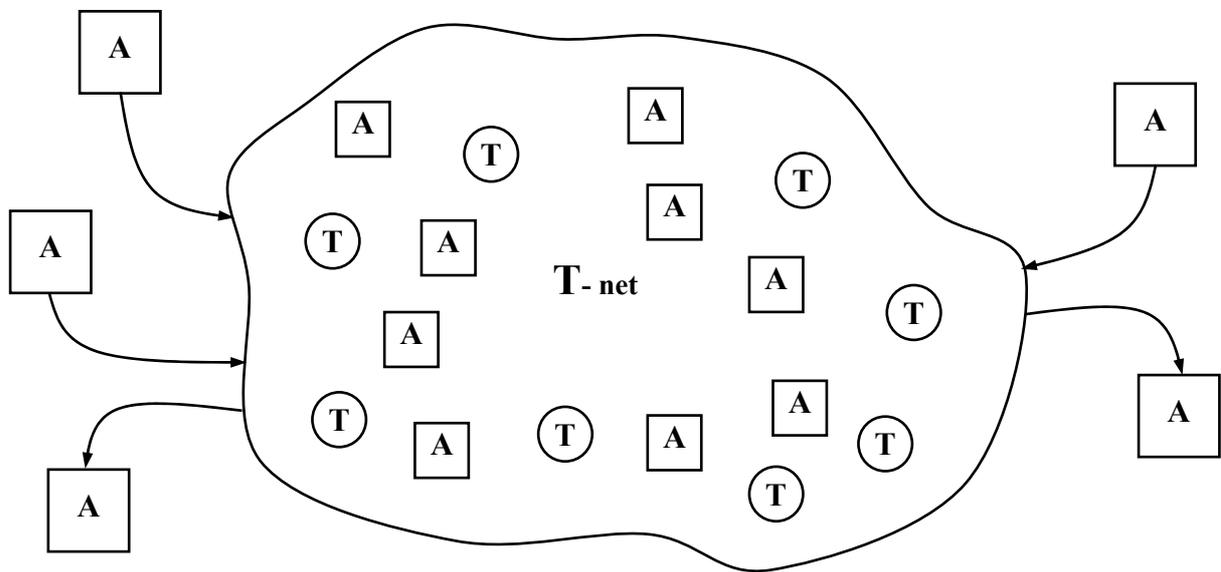


Figure 2d. Starting and finishing T-net accumulators
(source: self study)

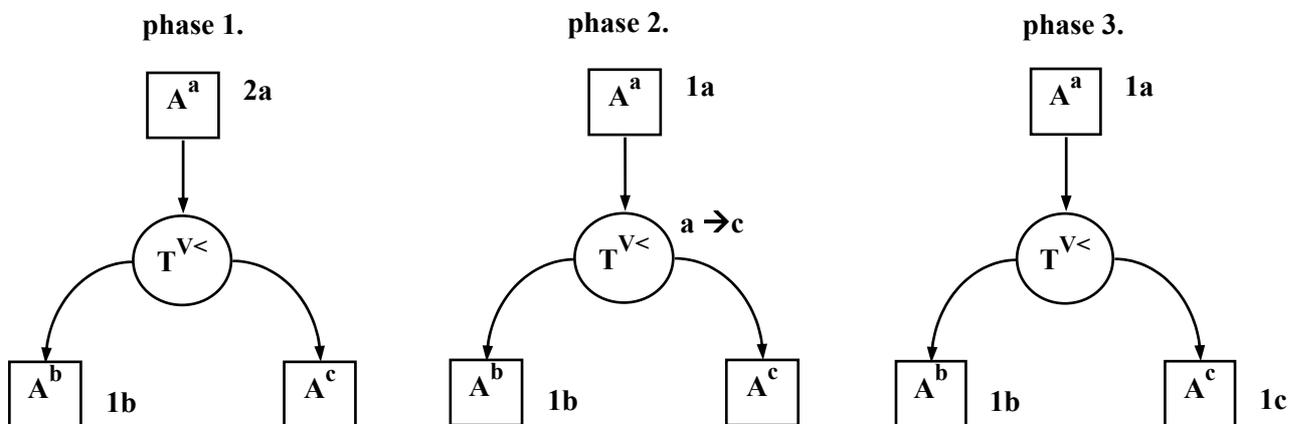


Figure 2e. $T_j^{V<}$ alternative dispersing transformer marker flow phases
(source: self study)

The main task of parallel dispersing transformer $T_j^{\&<}$ is to take one marker from the connected accumulator (input marker) and transforming it into a series of single outgoing markers and simultaneously, after certain τ_j time delay, transfer it through all parallel outputs of $T^{\&<}$ transformer one by one to a connected accumulator. Process flow is presented in the Figure 2g.

The main task of parallel gathering transformer $T_j^{\&>}$ is to take one input marker from all connected accumulators

and transforming it into a single outgoing marker and simultaneously, after certain τ_j time delay, transfer it through a single output of $T_j^{\&>}$ transformer to connected accumulator. Process flow is presented in the Figure 2h.

Transformer with only one input and one output section is called the basic transformer and marked as T_j without the distinguishing of its type (see Figure 2i).

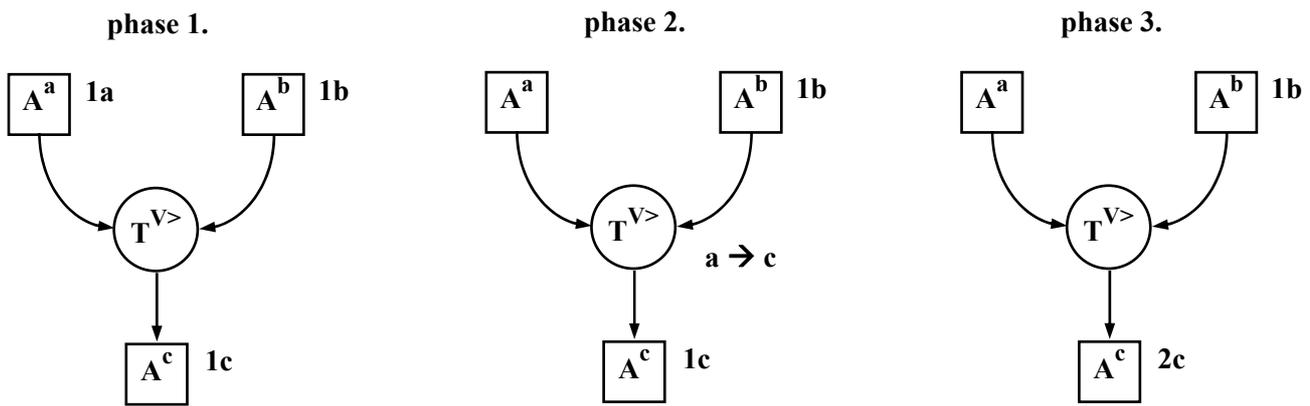


Figure 2f. $T_j^{V>}$ alternative gathering transformer marker flow phases
(source: self study)

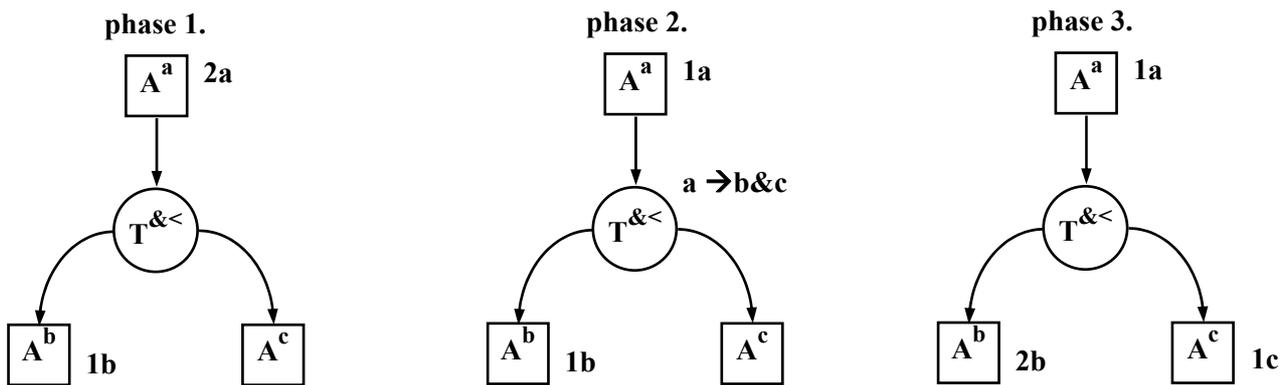


Figure 2g. Marker flow phases through dispersing parallel transformer $T_j^{\&<}$
(source: self study)

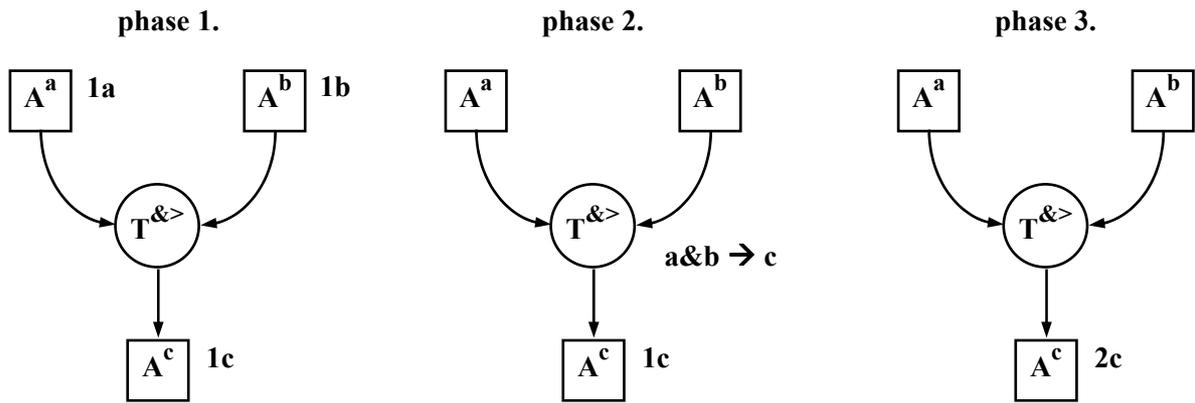


Figure 2h. Marker flow phases through gathering parallel transformer $T_j^{>}$ (source: self study)

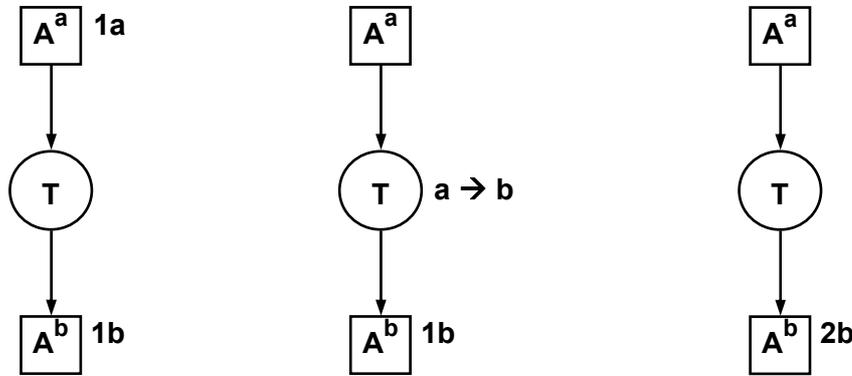


Figure 2i. Basic transformer (source: self study)

3 Transforming net functioning

Transforming net functioning can be described with a collection of internal transforming functions $\{T^{int}\}$ (see point 2).

Transforming net is functioning in an asynchronous manner due to the initiation of transformers in specific periods of time. Transformer can be in out of two allowable states:

- I standby, when:
 - transformer is not processing any incoming event marker sequence due to redundancy of its output accumulators what blocks the intake of markers (see Figure 3a); (maximum capacity of the accumulator is marked as $\max |A|$),
 - transformer is not processing incoming event marker sequence due to redundancy of input accumulators what prevents T^{int} projection from realization (see Figure 3b),

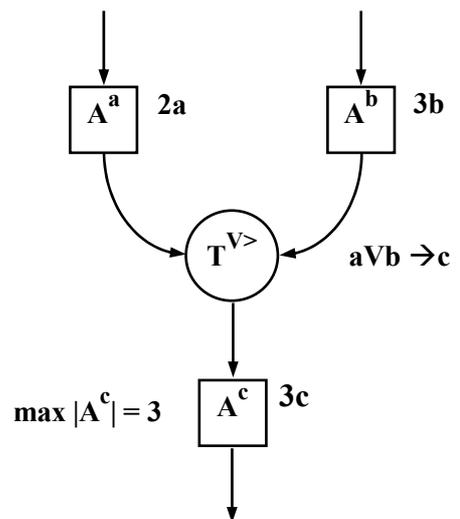


Figure 3a. Transformer in standby phase due to redundancy of the output accumulator (source: self study)

- II during transformation, which
 - starts from the selection of T^{int} projection (phase 1.), for which an event sequence, fulfilling this projection, will be found in input accumulators and simultaneously a space will be provided to take event markers, created as a result of chosen projection realization, in output accumulators (see Figure 3c),
 - afterwards event marker selection, from suitable input accumulators, is performed (phase 2.); these markers are, in certain τ time period, transformed according to T^{int} projection into event markers that suit the accumulators in the outgoing arcs (see Figure 3c),
 - it ends with the transfer of produced event markers (phase 3.) to the output accumulators (see Figure 3c).

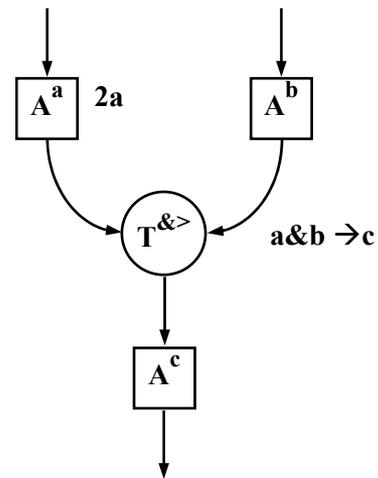


Figure 3b. Transformer in standby phase due to lack of required markers in input accumulators
(source: self study)

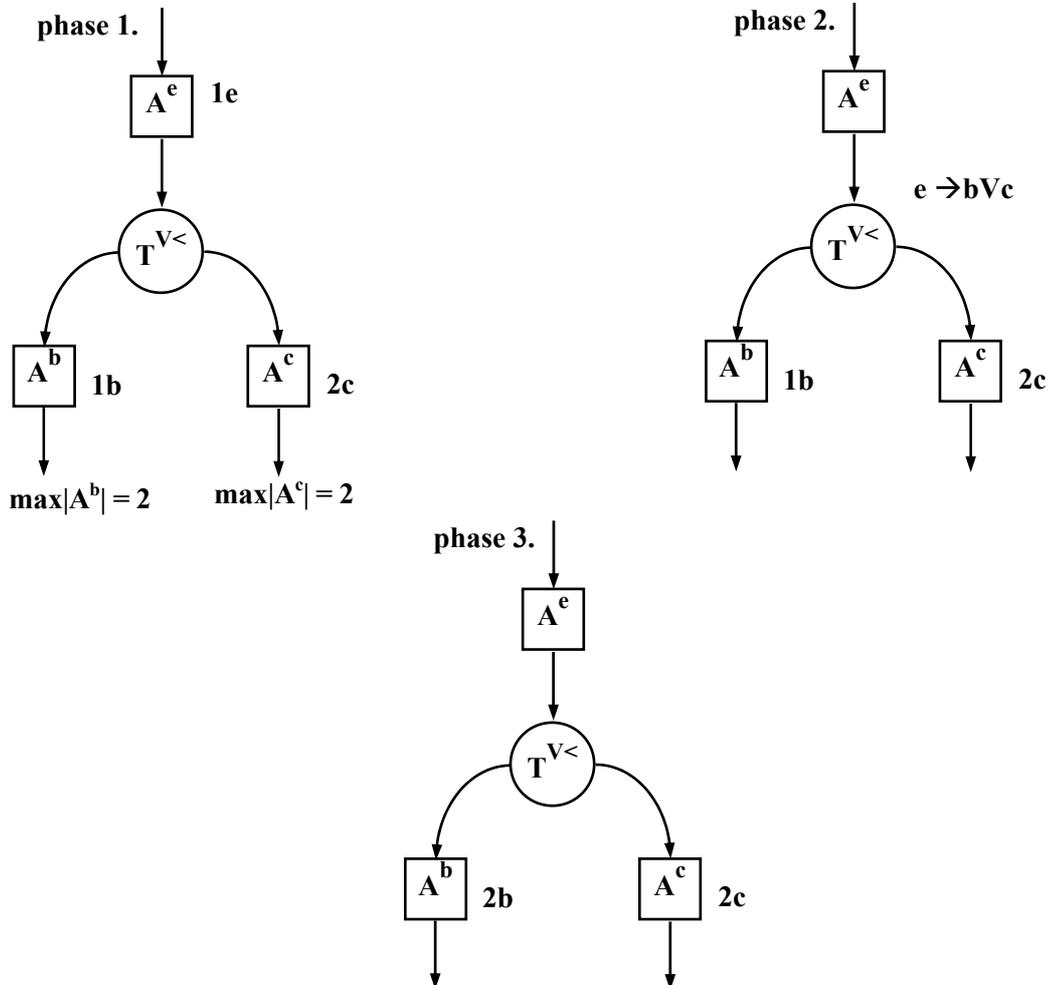


Figure 3c. Example of event markers transformation
(source: self study)

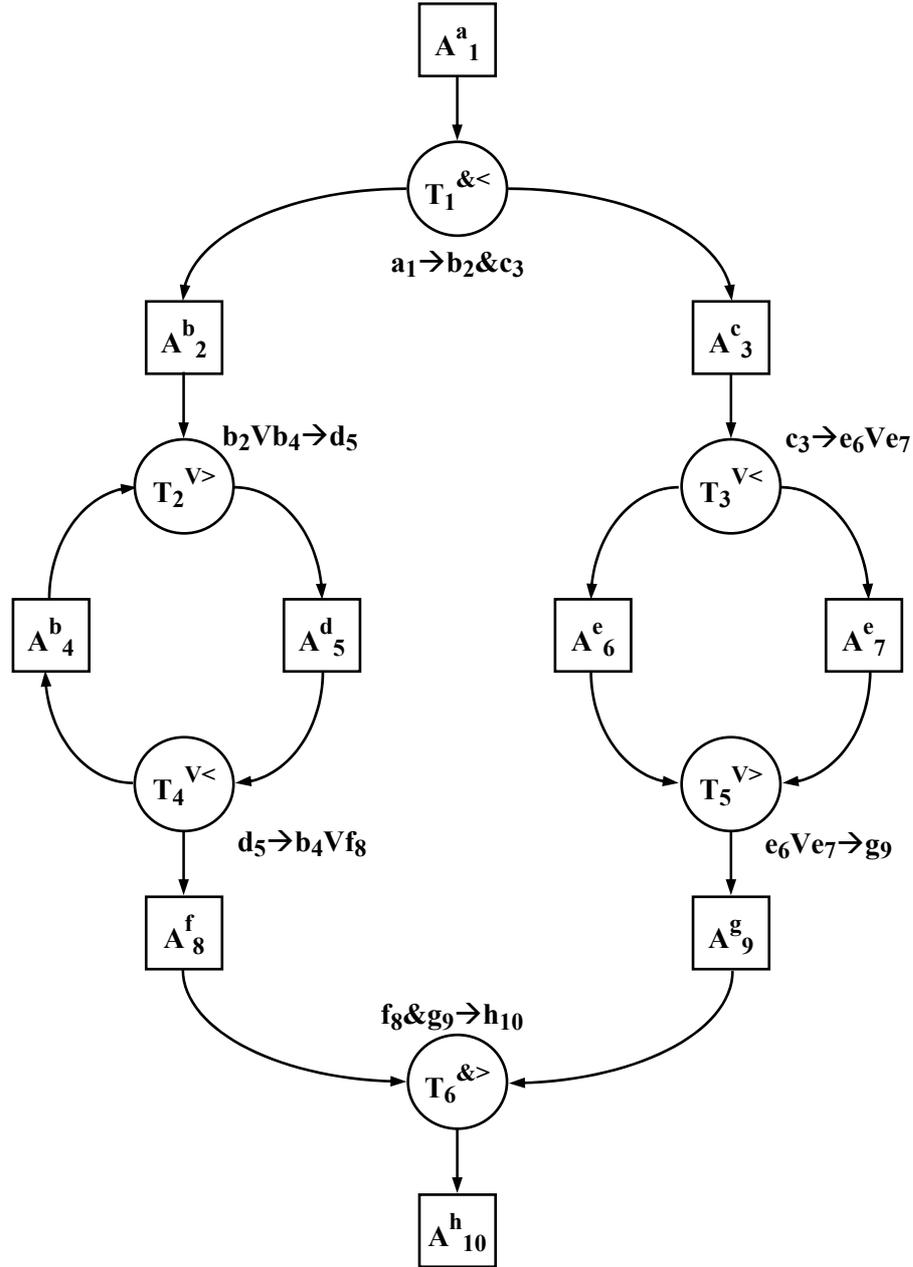


Figure 3d. T-net functioning example (source: self study)

Example of a T-net functioning, consisting of six $T_1 - T_6$ transformers and ten $A_1 - A_{10}$ accumulators, is presented in the Figure 3d:

- $T_1^{&<}$ transformer (see Figure 3d) realizes the internal transforming function T_1^{int} , in the form of a projection: $a_1 \rightarrow b_2 \& c_3$ of type a markers collected from A^a_1 accumulator and changed for type b and c markers transferred to A^b_2 and A^c_3 accumulators,
- $T_2^{V>}$ transformer realizes the internal transforming function T_2^{int} , in the form of a projections: $b_4 \vee b_2 \rightarrow d_5$ of type b markers collected from A^b_2 or A^b_4

accumulators and changed for type d markers transferred to A^d_5 accumulator,

- $T_3^{V<}$ transformer realizes the internal transforming function T_3^{int} , in the form of a projection: $c_3 \rightarrow e_6 \vee e_7$ of type c markers collected from A^c_3 accumulator and changed for type e markers transferred to A^e_6 and A^e_7 accumulators,
- $T_4^{V<}$ transformer realizes the internal transforming function T_4^{int} , in the form of a projection: $d_5 \rightarrow b_4 \vee f_8$ of type d markers collected from A^d_5 accumulator and changed for type b and f markers transferred to A^b_4 and A^f_8 accumulators,

- $T_5^{V>}$ transformer realizes the internal transforming function T_5^{int} , in the form of a projections: $e_6 \vee e_7 \rightarrow g_9$ of type e markers collected from A_6^e or A_7^e accumulators and changed for type g markers transferred to A_9^g accumulator,
- $T_6^{\&>}$ transformer realizes the internal transforming function T_6^{int} , in the form of a projections: $f_8 \& g_9 \rightarrow h_{10}$ of type f and g markers collected from A_8^f and accumulator and changed for type h markers transferred to A_{10}^h accumulator.

4 Colored and transforming Petri net

Transforming net functioning can be presented in the representation of colored Petri net.

Figure 4a presents a situation where Petri net transition plays a role of a conjunctive connection from the incoming side. In phase 2., activation of t_1 transition with

z marker takes place; in T-nets $z\&z \rightarrow z$ projection realization corresponds with this event.

All considered examples of event modeling in Petri net with T-nets relate to the situation where the internal transforming function T^{int} describes x marker projections of different types (colors); markers of the same color are always present in both of its sides.

Figure 4b presents a situation where Petri net transitions are in conflict for two different markers b and z . In this situation p_1 position plays the role of the alternative connection from the output side. Conflict resolution can happen in two “symmetrical” ways – after phase 1., accordingly phases 2, 2’ as well as 3 and 3’ take place. Special attention should be drawn to the fact that two Petri net transitions are modeled with one transformer $T^{V<}$, which can process b and z markers only in sequential manner what as a result causes the process to be prolonged by one time step (phase 4).

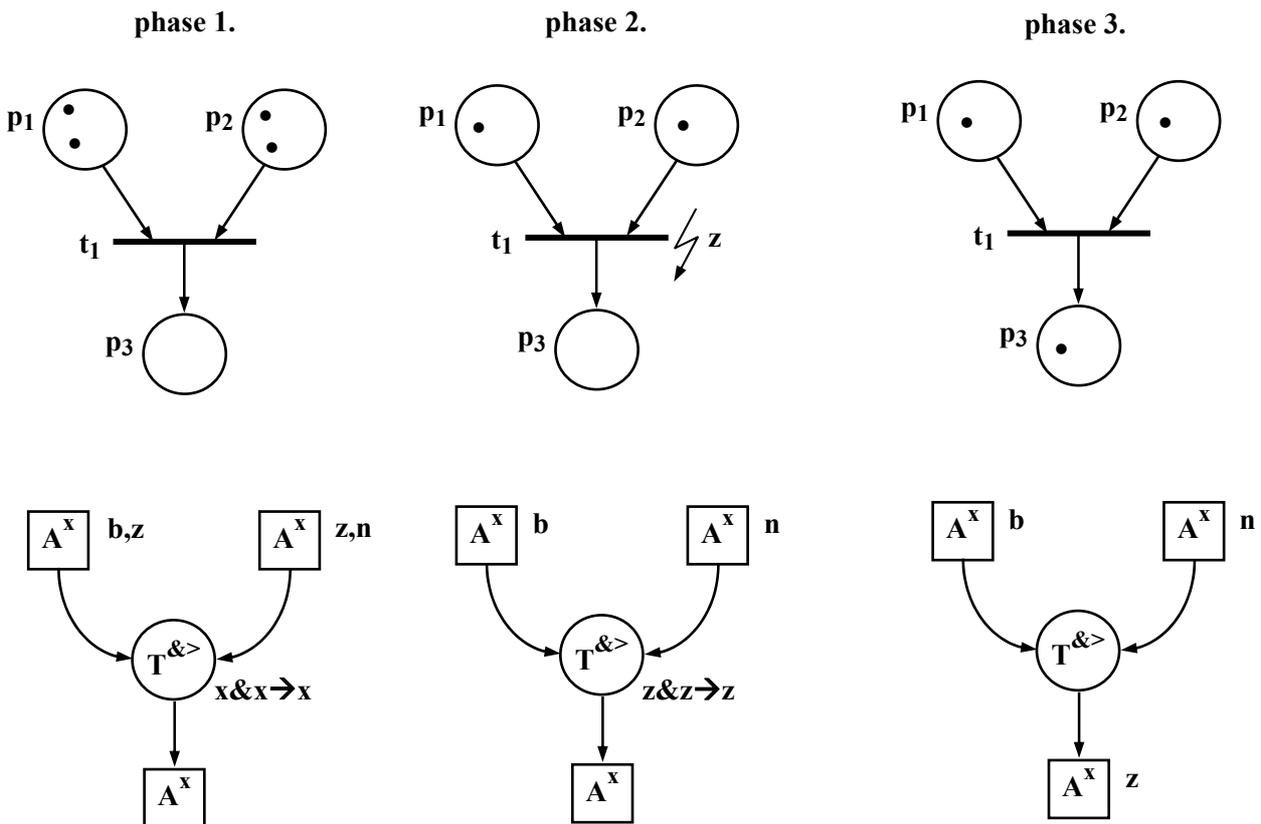


Figure 4a. Examples of marker transformation phases in Petri net and in T-net (in this case: t_1 transition plays the role of conjunctive connection from the incoming side).
(source: self study)

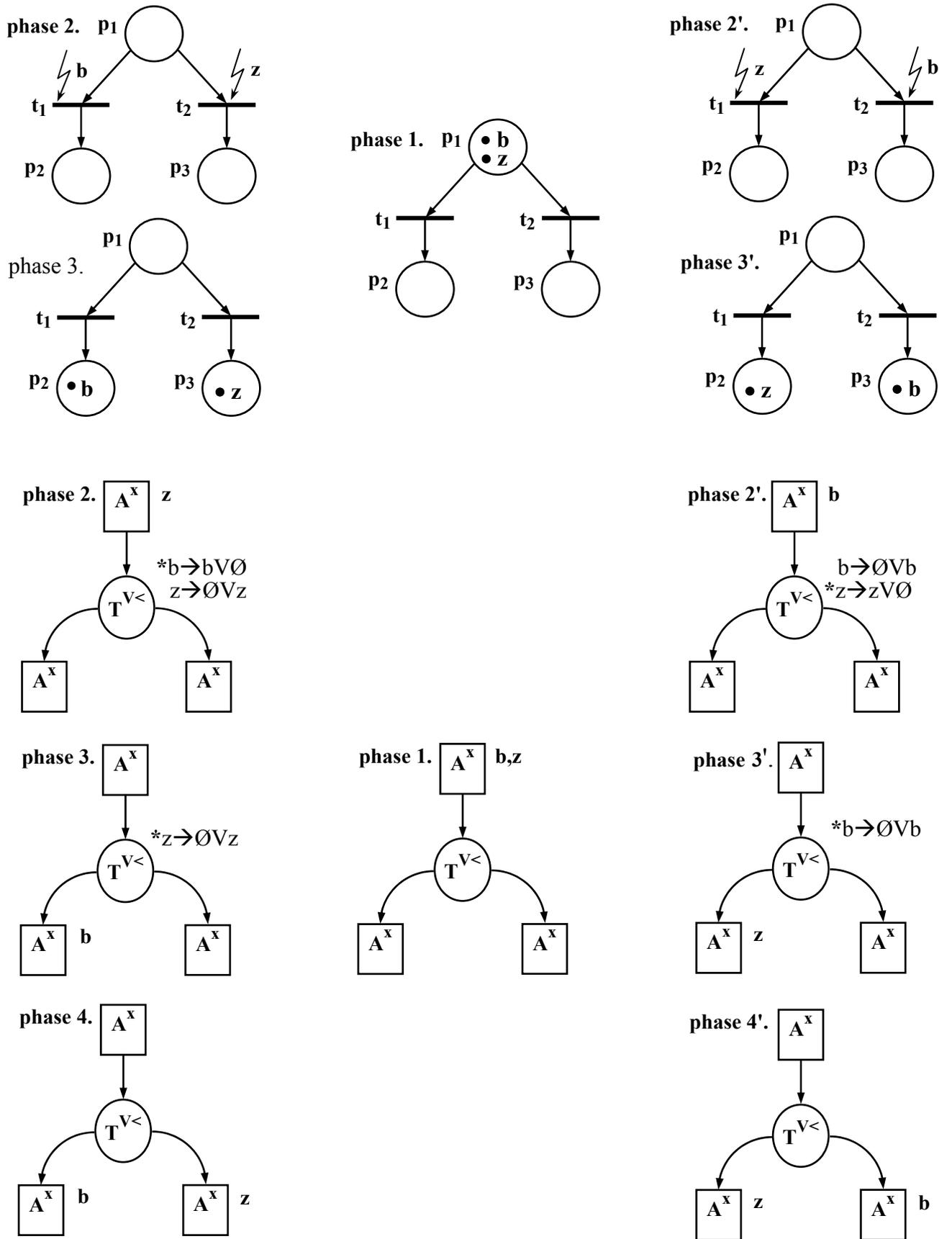


Figure 4b. Examples of marker transformation phases in Petri net and in T-net
 (in this case: p_1 position plays the role of alternative connection from the outgoing side)
 (source: self study)

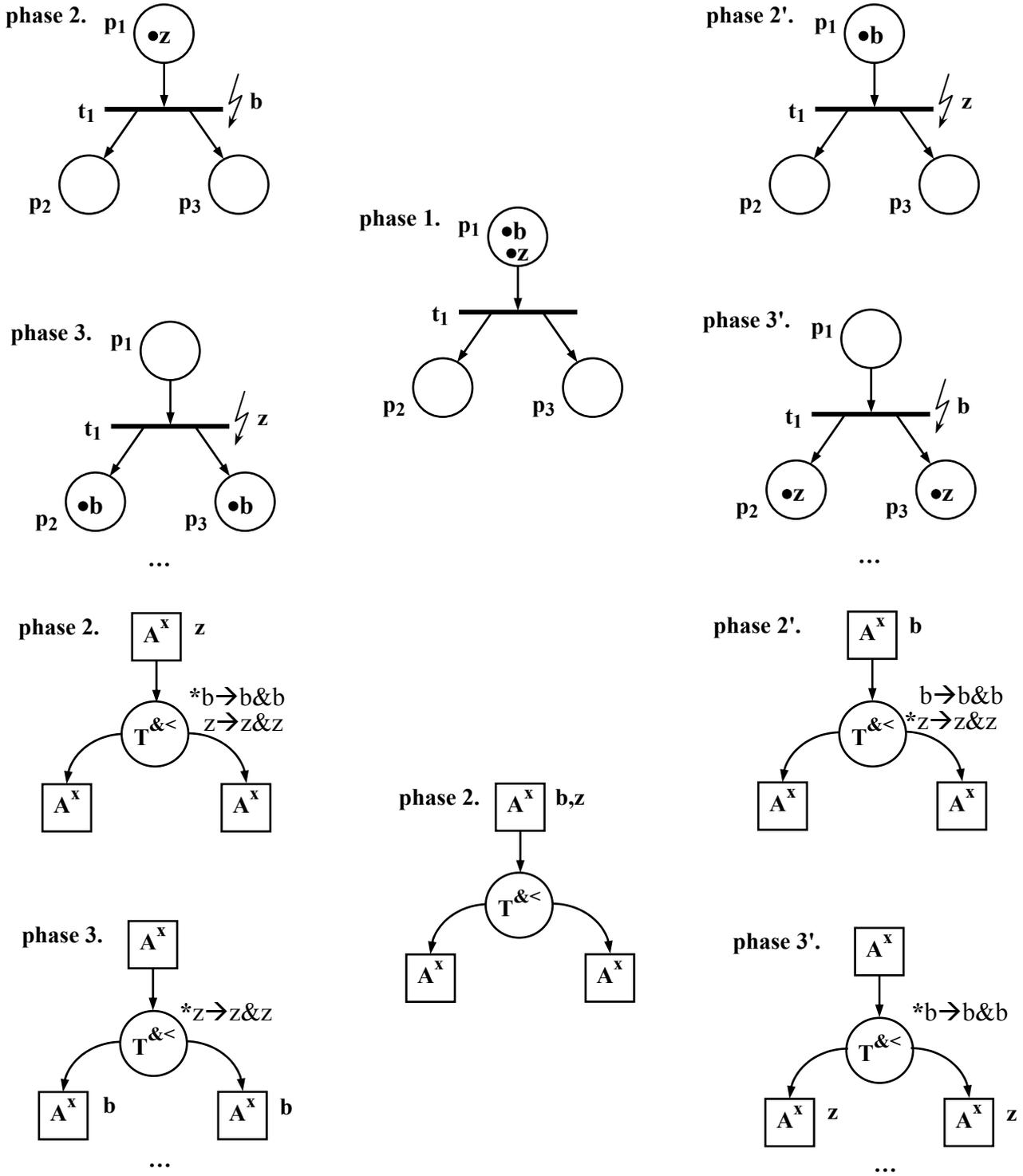


Figure 4c. Examples of marker transformation phases in Petri net and in T-net (in this case: t_1 transition plays the role of conjunctive connection from the outgoing side) (source: self study)

Figure 4c presents a situation where the Petri net transition plays the role of conjunctive connection from the outgoing side. Finally, in the following phase 4., markers b, z appear at the p_2 and p_3 positions. In this

case both t_1 transition and corresponding $T^{\&<}$ transformer are functioning similarly and sequentially transforming the markers.

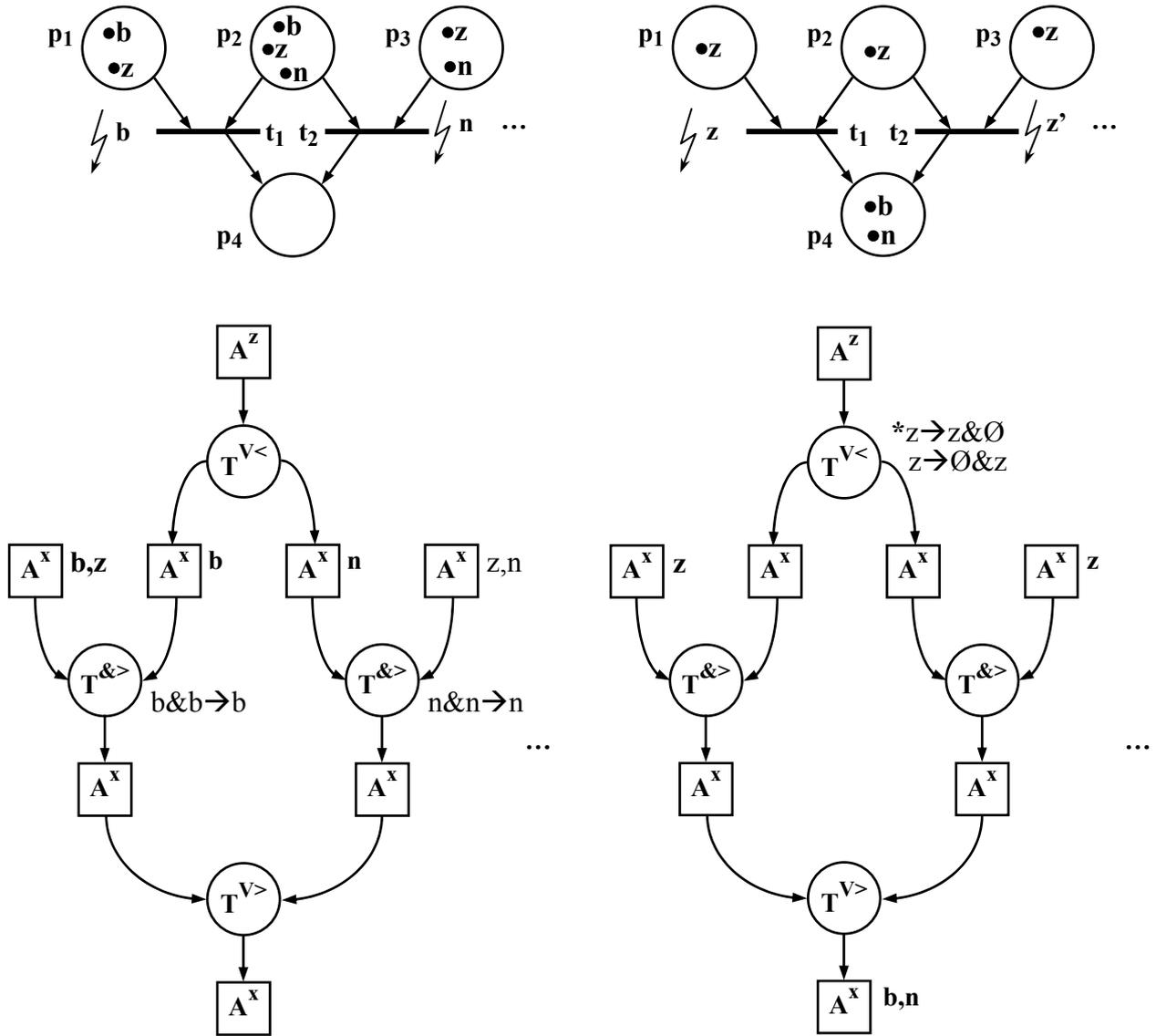


Figure 4d. Examples of marker transformation phases in Petri net and in T-net
 (in this case: p_2 position supplies two transitions – in T-nets buffer can only supply one transformer)
 (source: self study)

Figure 4d presents a situation where p_2 position as well as t_1 and t_2 Petri net transitions create a configuration that is not allowable for direct depiction and T-net symbolism, due to the fact that one transforming net accumulator can be connected to only one transformer from input and output side. Similar situation can be found in a Petri net where, for example, p_2 and p_4 positions were connected to two transitions (there is no limitation for the number of mutually connected transitions and positions). In given example the T-net is two times bigger than the modeled Petri net, whereas there are no situations (conflicts) present that would require suitable solving mechanisms.

5 Multitask transforming net

Multitasking is one of the natural features of transforming net. One can interpret a single processed marker as a task, in order to test this feature. It is necessary to assume at this point that tasks can be subjected to a multilevel decomposition (disaggregation) until the basic tasks layer and its results can be merged (aggregated) in the opposite direction (from elementary resources tasks to complex resources tasks).

Possibility to time-flow tasks is immanent features of T-nets – resources (their markers to be precise) are gathered in the accumulators; resource (marker) transformations are described by the transformers. Interpretation of desirable (expected) resources (markers) in accumulators is a remaining problem.

The essence of multitasking, in T-net interpretation, is based on such planning (addressing in time to proper accumulators) and coordination (priorities of task markers in accumulators, projection variants in transformers consistent with marker priorities) of task flow (according to Figure 1a) that is realized according to the aim function expressing the commitment of natural or automatic “system administrator”.

The way the aim function is “translated” into decomposition and task flow placement in T-net accumulators is not the subject of this elaboration at this point.

T-net multitasking will be considered with the assumption that its structure was designed to perform specific task class, which unambiguously determine the sub-collection of output accumulators (final tasks) – and

simultaneously that realization of tasks assigned to input accumulators (cooperation tasks) for given net is assured (see Figure 2d).

Every T-net accumulator determines through its markers a single type of task consistent with the type of these markers. Many accumulators for the same type of task can be present in one net (processing the markers of the same type), powering different T-net transformers.

For example, net is in the form presented in the Figure 5a (see point 1). The X_P task starts the process of task realization and the X_Q task finishes it. Tasks do not have a dedicated space (infrastructure) for task realization. Tasks in the task net are connected by alternative and parallel connections. X_k and X_l tasks control the order of task selection with $A^<$ alternative dispersing connections.

Tasks are realized as predicate functions, which on the basis of task input data determine the order of following tasks (with the numbers 0 or 1), in order to make this process possible.

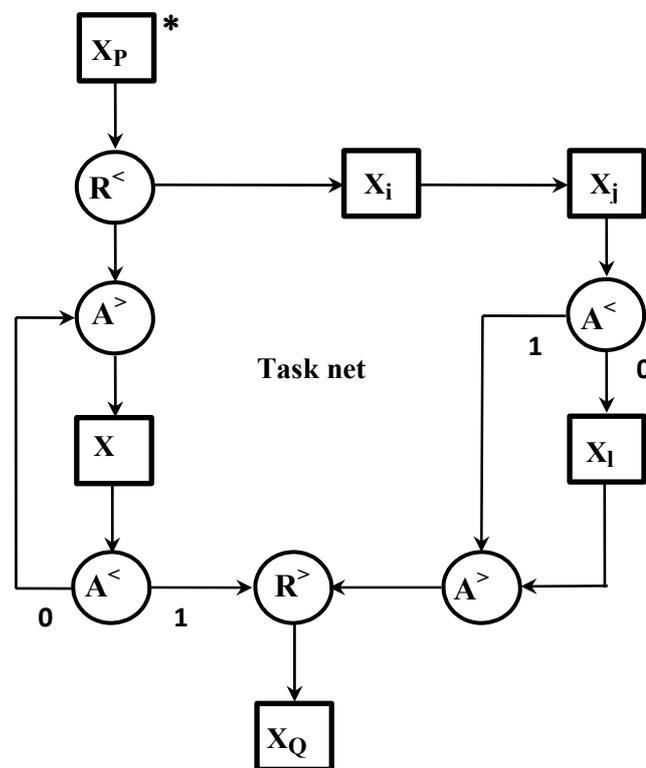


Figure 5a. Initial state of net tasks activity (* - active net element, X - task, $R^<$ - parallel dispersing connection, $R^>$ - parallel gathering connection, $A^<$ - alternative dispersing connection, $A^>$ - alternative gathering connection)

(source: self study)

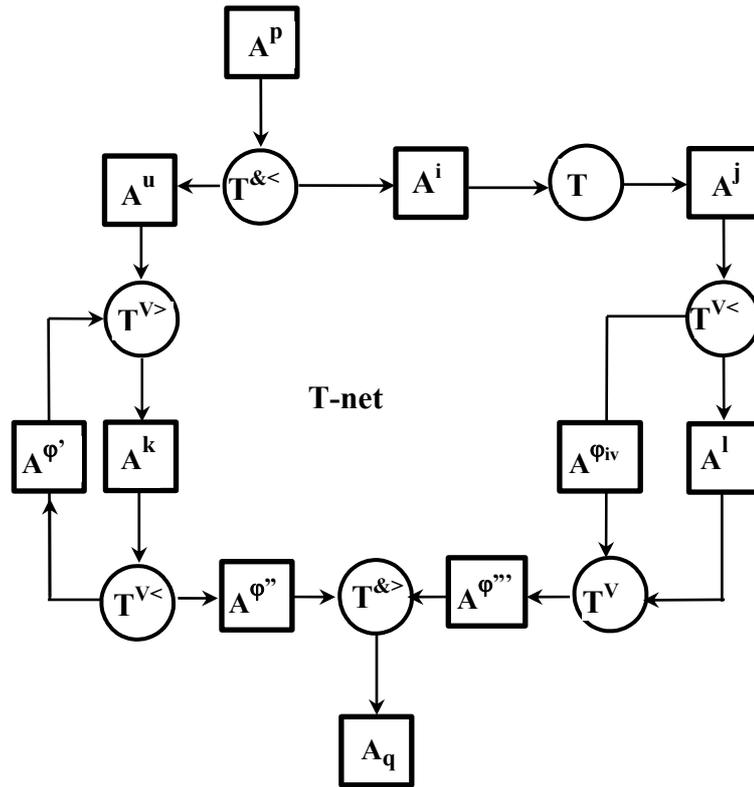


Figure 5b. T-net corresponding to task net (see Figure 5a) (source: self study)

Net presented in the Figure 5a does not “show” the method (technology) of task realization, there is no possibility to model task realization of $X_p, X_{p'}, X_{p''}, X_i, X_{i'}, X_j, X_{j'}, X_{j''}, \dots, X_Q, X_{Q'}, X_{Q''}$ stream.

It is necessary to prepare, for every task in the net, at least one accumulator, which can store its markers, in order to allow it. At the same time transformers for task nets, which will perform operations assigned to parallel and alternative connections in the task nets, need to be determined. Many different variants of a given task can be realized (performed) or set for realization (planned), as a part of particular task type, in a T-net accumulator. Therefore construction of multitask net needs to be predicated with the determination of task types that will be realized by given net. Suitable number of accumulators, with a capacity that will assure a harmonious task realization, needs to be selected for every defined task type.

T-net with one input and output accumulator realizing task net from Figure 5a is presented in the Figure 5b.

Due to the constraints present in T-nets (see Figure 5b) it is necessary to introduce additional elements (accumulators), which would divide the neighboring transformers, substituting for elements from the original net. Such

accumulators gather tasks present before and after the transformation.

Constructed T-net can serve many tasks in parallel (all of its transformers can be active simultaneously) and their markers are stored in accumulators. Accumulators can also realize the “policy” of task (marker) arrangement according to set priority system, what takes place in task realization planning and control system e.g. in real production systems.

Determination of T- net tasks is conditioned by its four transformers and realized transforming tasks functions $T^{&>int}, T^{&<int}, T^{V>int}$ and $T^{V<int}$ (see point 2).

6 Transforming net modeling

Transforming net (T-net) can be modeled with the use of techniques used in case of Petri nets (e.g. reachability graph) or through regular expressions used in event nets [11, 14].

Essential practical aspect of every network structure is its openness understood as the possibility to attach new or close down (eliminate) existing elements. In case of T-nets it usually means connecting or eliminating accumulators and transformers.

The most basic T-net (see Figure 2i) consists of two accumulators and a connecting transformer. Such net realizes operations based on tasks (or its markers) described with a T transforming function.

Development of the net presented in the Figure 2i can be realized through:

- adding accumulators:
 - any amount of input accumulators with a simultaneous declaration of transformer type (alternative or parallel gathering transformer); see Figure 6a,
 - or any amount of output accumulators with a simultaneous declaration of transformer type (alternative or parallel dispersing transformer); see Figure 6b,
- adding transformers:
 - adding one alternative transformer or parallel gathering from the side of input accumulator; see Figure 6c,
 - adding one alternative transformer or parallel dispersing from the side of output accumulator; see Figure 6c,
- transformer decomposition:
 - into a pair of alternative transformers connecting any number of accumulators; see Figure 6d,
 - into a pair of gathering transformers connecting any number of accumulators; see Figure 6d.

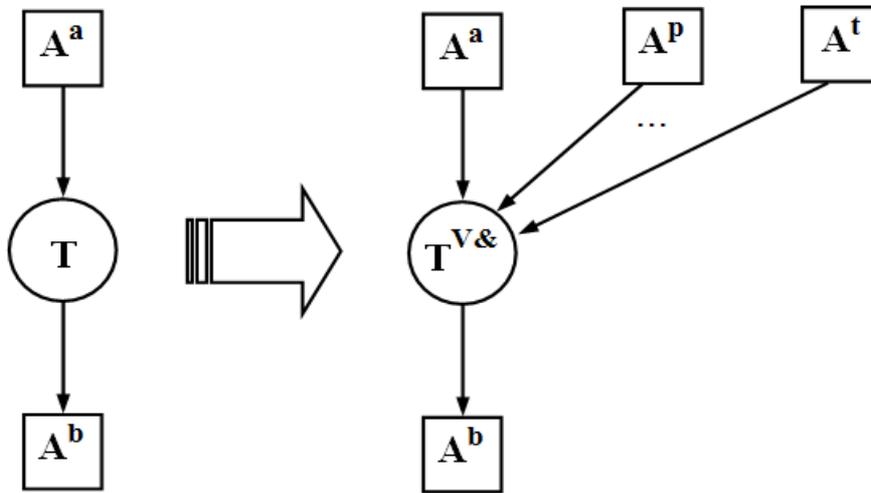


Figure 6a. Adding input accumulators
(source: self study)

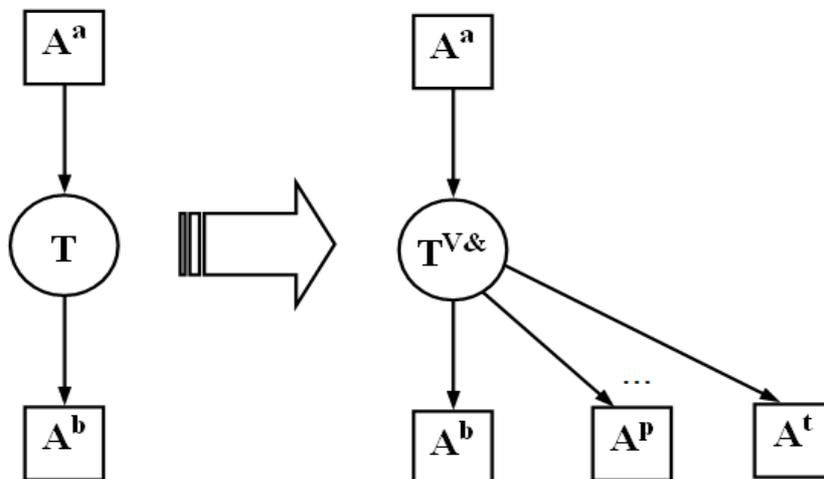


Figure 6b. Adding output accumulators
(source: self study)

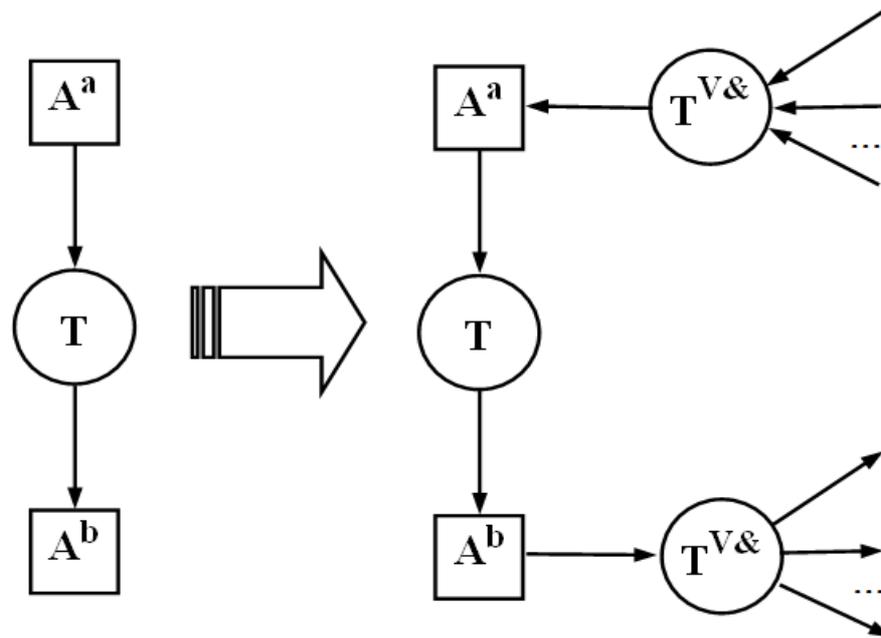


Figure 6c. Adding transformers (*source: self study*)

Presented examples of T-net development do not exploit all network development possibilities. This is especially significant for T-nets with periodically connected accumulators and transformers (see Figure 5c).

It is necessary to emphasize that adding a new transformer into a T-net requires a simultaneous adding of at least two accumulators (one from input and one from output side).

It is crucial to ask about the possibility to stimulate the activities of a T-net in a situation when it realizes the processing of task flows randomly appearing at its input accumulators. Such network should assure realization of incoming tasks within the set time and cost limitations.

If one assumes that T-net accumulators have limited capacity of stored task markers and that transformers simultaneously realize a limited number of allowable alternative transforming functions in a limited time, these constraints condition the limited of T-net states - where a T-net state is understood as a current placement of markers in all accumulators with simultaneously realized collection of transforming functions.

It is easy to notice that even in case of relatively small T-nets the number of its allowable states grows exponentially in relation to the number of accumulators, its capacity, time characteristics and transformer

functions. For example a T-net shown in the Figure 3c, with an assumption that every transformer can have no more than two markers, can have the estimated number of states at the level of 16 thousand without the inclusion of the transformer function realization time.

The important aim of construction and modeling of different T-net variants is the analysis of its correctness according to set system of correctness axioms. These axioms should be determined a priori – before the modeling starts.

Axiom defined net functioning correctness is the first step to prove the correctness of any defined network if one can prove that the net fulfills correctness axioms – it is correct according to the axioms. Axiom correctness proof cannot, however, be a result of tests. It is necessary use the theory of system characterization, formulated by V.A. Gorbatow and confirmed in many elaborations on characterization theory [1-3 and 13].

Network correctness axiom definition needs to refer to its structure features, especially for alternative and parallel transformers, which structure features (number of inputs and outputs) should be in accordance with its technological formulas. In such case a local correctness, for every transformer in relation to connected accumulators, can be defined.

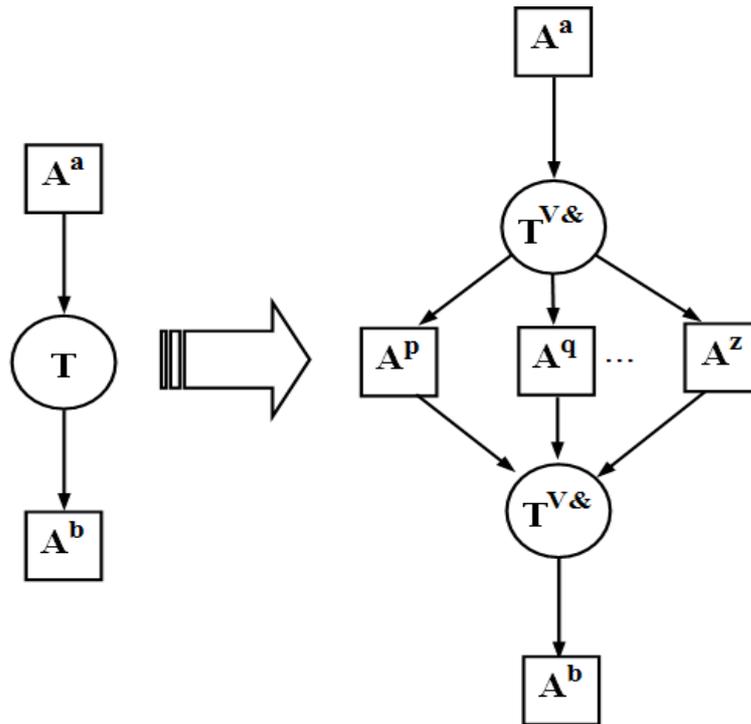


Figure 6d. Transformer decomposition into pair of alternative transformers or pair of parallel transformer
(source: self study)

It is essential to bear in mind that local correctness is a necessary condition for the functioning of the whole network (global correctness), which is not equal to the sum of local correctness and must be defined in a way that suits the constraints of starting and finishing accumulators in the network (see Figure 2d).

In every case the main task of the T-net task modeling and simulation is the construction of a priori flawless structures or the search the search of errors in existing nets through attempts to prove its correctness.

In case of T-nets a calculation effective solution can be assuming the constructive procedure of correctness assurance or the approach based on the characterization theory.

7 Transforming network design

T-net design should lead to the accomplishment of network that fulfills functional requirements – realizes specific class of tasks in a correct manner (according to given constraints). Therefore the first step is to determine task class for the network to realize. Despite of the obviousness of this statement it is not that simple to fulfill it.

Usually the definition of a product structure in manufacturing activities is not directly connected with the manu-

facturing technology. Similar situation can be found in service processes, which can be realized in many different manners.

One of the practical issues in effective production and service task realization is the determination of activities necessary for task decomposition into more basic units and the other way around – synthesis of basic tasks results into the final task (or the primary task that was the subject of decomposition).

Task decomposition and task result synthesis is performed in a continuous manner. The main reason for a single task decomposition is to bring it from the planning to operational level (more detailed), which will gather all necessary components that will be later merged (composed) and will provide a result suitable for the final task.

During the procedure of transforming network design one needs to assume that every formulated task will have one corresponding resource with two distinguished states:

- R_z resource distinguished state preceding the task formulation,
- R'_z distinguished state corresponding to the standby task realization.

Task realization can have different results and in boundary cases can lead to partial realization or even cancellation. Every resource participating in task realization, stored in a particular type of T-net accumulator, has its own set of features and its values. Set collection of features determines the class of the resource.

Set collection of resource features values for given class distinguishes resource type of certain class. T-net transformers task is to:

- decomposition of a complex resource from given class and type into partials from other classes till the class and type of elementary resources, which are not subjected to further decomposition; in special cases of the decomposition process there can be only one partial resource – in such case both resources will belong from different classes and different resource types,
- partial resources synthesis, from different classes and types, to a complex resource with a simultaneous determination of resource feature value (setting the type of the complex resource); in special cases the synthesis process, when the feature list is changed, from one partial resource only one complex resource can be created – in this case (similarly to the decomposition process) both resources will belong to different classes and different types of resources,
- change of given class resource type into a new and desired resource of the same class as a result of given type output resource feature values transformation.

T-net design process should be started from the design of type trajectory for the resources of all classes. Type trajectories are the illustration of changes of given resource in technological processes. Properly designed T-net connects decomposition, synthesis and change resource transformers.

8 Summary

The material presents verified concept of T-nets [4]. Verification was based in the classic functionality analysis of Parallel Block Schemes (PBS) [1, 3], colored Petri nets and first models of transforming nets constructed on the basis of PBS schematics [8, 9, 11, 13 and 14].

T-net is build in a manner that connects the technological resource processing aspects with the logical requirements for the order, time and cost of realization of technological operations. T-net structure allows parallel realization

of many tasks simultaneously with the consideration of priorities resulting from T-net state monitoring.

Correctness of networks is defined with properly determined axioms, which can be “weakened” or detailed in relation to current T-net state. In each case local and global T-net task realization must be assured.

T-net plays the role of virtual model of real task realization processes. The existence of such model allows to monitor and asses the progress of realized tasks as well as calculations and simulation of task realization strategy.

Proposed T-net theory elements allow the verification of proper functioning axioms without the necessity of testing of network and its elements configuration.

Many of described issues and solutions still have conceptual character and is subjected to verification and change.

9 References

- [1] Kim D.P., Nazaretov V.M., Krupa T. - *Techniczna imitacja intelektu*. WNT, Warszawa 1991.
- [2] Krupa T. - *Częściowo uporządkowane systemy V.A. Gorbatowa* [in] Przegląd Organizacji, No. 12, 1976 Warszawa, pp. 541-543.
- [3] Krupa T. - *Zasada charakteryzacji w projektowaniu systemów organizacyjno-technicznych* [in] Zeszyty Naukowe IOZ PW - Organizacja i Zarządzanie Przemysłem. Oficyna Wydawnicza PW, No. 1, Warszawa 1983, pp. 33-51.
- [4] Krupa T. - *Parallel Processing of Resources in T-networks* [in] Int. Conf. SYPRO'90 Systemy Produkcyjne - Teoretyczne i Praktyczne Problemy Projektowania. IOSP PW, Warszawa 1990, pp. 443-449.
- [5] Krupa T., Ostrowska T. - *The Logical Aspects of Tasks and Resource Design* [in] Int. Conf. Cybernetics and Systems'90. World Scientific, University of Vienna, Vienna 1990, pp. 267-274.
- [6] Krupa T. - *Zadaniowy model sterowania produkcją w warunkach komputerowej integracji przedsiębiorstwa CIM-ARIS Computer Integrated Manufacturing* (ed. T. Kasprzak). Wydział Nauk Ekonomicznych, Uniwersytet Warszawski 1995, pp. 90-116.
- [7] Krupa T. - *Problemy projektowe systemów informatycznych zarządzania* [in] Zeszyty Naukowe IOSP PW - Organizacja i Zarządzanie Przemysłem. Oficyna Wydawnicza PW, No. 10, Warszawa 1998, pp. 99-127.

- [8] Krupa T. - *Wielozadaniowa sieć transformująca* [in] *Komputerowo Zintegrowane Zarządzanie* (ed. R. Knosala). WNT, Warszawa 2000, pp. 326-338.
- [9] Krupa T. (ed.) - *New Challenges and Old Problems in Enterprise Management*. WNT, Warszawa 2002.
- [10] Krupa T. - *Concept of Organizational Change* [in] *New Challenges and Old Problems in Enterprise Management* (ed. T. Krupa). WNT, Warszawa 2002, pp. 229-252.
- [11] Krupa T. - *Kolorowana sieć Petri'ego i sieć transformująca* [in] *Zeszyty Naukowe IOSP PW - Organizacja i Zarządzanie Przemysłem*. Oficyna Wydawnicza PW, No. 15, Warszawa 2002, pp. 5-24.
- [12] Krupa T. - *O pewnej specyfikacji procesu decyzyjnego* [in] *Problemy Zarządzania*. Wydawnictwa Naukowe WZ UW, No. 1, 2005, pp. 79-88.
- [13] Krupa T. - *Elementy organizacji. Zasoby i zadania*. WNT, Warszawa 2006.
- [14] Krupa T. - *Sieciowe modele procesów zdarzeń* [in] *Komputerowo zintegrowane zarządzanie* (ed. R. Knosala). Oficyna Wydawnicza PTZP, Opole 2006, pp. 55-66.