

Now we take a little closer look at Petri nets, that is, at their structure of places, transitions and arcs, the fundamental data structure of multisets, the structure of markings and steps and lastly the reachable markings and the final markings. We explain this with the help of the (slightly modified) cookie vending machine.

2.1 A Variant of the Cookie Vending Machine

Figure 2.1 shows a modified version of the cookie vending machine previously shown in Fig. 1.10 (the denotations A ... H of the places and a ... e of the transitions make the notation easier). In addition to the five rectangular cookie packets,

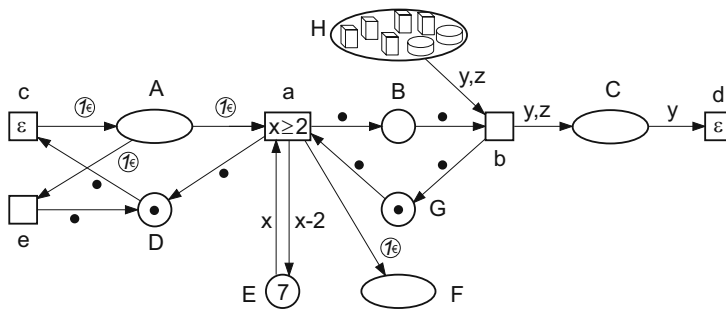


Figure 2.1 Two kinds of packets and giving out two packets at once

two round packets are now in the storage H. The customer receives *two* cookie packets for one euro. The machine decides non-deterministically whether those packets are rectangular or round. Bought packets are dropped into the compartment C. The customer can remove one packet at a time (via the cold transi-

two kinds of tokens:



tion d). The net in Fig 2.1 will be used as an example throughout Chap. 2.

2.2 Components of a Net

The example of the cookie vending machine shows all the kinds of components that can occur in a Petri net.¹ We will look at them again individually and explain their roles in the model of the system.

Places



A Petri net is a structure with two kinds of elements. One kind of element is *places*. Graphically, a place is represented by a circle or ellipse. A place p always models a *passive* component: p can store, accumulate or show things. A place has discrete states.

Transitions



The second kind of elements of a Petri net are *transitions*. Graphically, a transition is represented by a square or rectangle. A transition t always models an *active* component: t can produce things, consume, transport or change them.

Arcs



Places and transitions are connected to each other by directed *arcs*. Graphically, an arc is represented by an arrow. An arc never models a system component, but an abstract, sometimes only notional relation between components such as logical connections, access rights, spatial proximities or immediate linkings.

In the example of the cookie vending machine, it is striking that an arc never connects two places or two transitions. An arc rather runs from a place to a transition or vice versa from a transition to a place. This is neither coincidental nor arbitrary, but

¹ The literature gives a multitude of extensions and generalizations, which are not covered here.

inevitably follows if nets are used correctly to model systems, that is, if passive and active components are properly separated.

Net Structure

It is customary to denote the sets of places, transitions and arcs with P , T and F , respectively, and to regard arcs as pairs, that is, F as a relation $F \subseteq (P \times T) \cup (T \times P)$. Then

$$N = (P, T, F)$$

is a *net structure*. The places and transitions are the *elements* of N . F is the *flow relation* of N . Figure 2.2 shows the net structure of the cookie vending machine as shown in Figs. 1.10, 1.11 and 2.1.

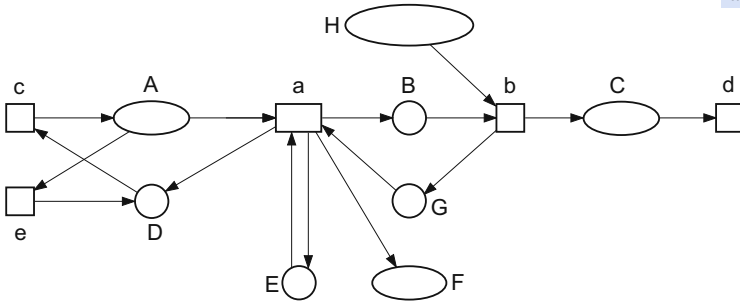


Figure 2.2 The net structure of the cookie vending machine

If a given context unambiguously identifies a net N , the *pre-set* $\bullet x$ and *post-set* x^\bullet of an element x are defined as

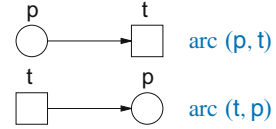
$$\bullet x =_{\text{def}} \{y \mid yFx\} \quad \text{and}$$

$$x^\bullet =_{\text{def}} \{y \mid xFy\}.$$

Two elements x, y of N form a *loop* if $x \in \bullet y$ and $y \in \bullet x$. For instance, a and E in Fig. 2.2 form a loop.

Markings

A *marking* is a distribution of tokens across places. A marking can be represented graphically by symbols serving as tokens in

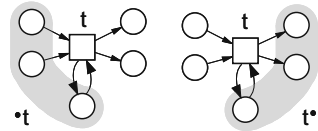


net structure²

place

transition

arc



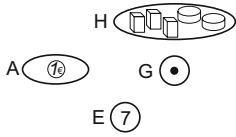
pre-set $\bullet x$

post-set x^\bullet

loop

² A highlighted term in the margin, like this one, refers to a definition in the 'Formal Framework' (see p. 213).

tokens in places:

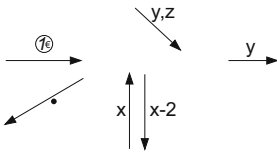


the respective circles and ellipses. For a system with an initial state, the initial marking is often depicted in this way. The symbolic tokens (for instance, 10, 100, 7) generally denote elements of the real world. This correlation is so strong that we do not distinguish between the symbolic representation and the real elements that they denote.

Next to symbolic tokens, abstract black tokens often occur, for instance, in the places D and G in Fig. 2.1. Such a token often indicates that a certain condition (modeled as a place) is met. It is also possible, and common, to represent concrete elements not by symbolic but by abstract black tokens. *Elementary* nets only utilize black tokens.

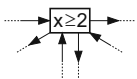
Labelings of Arcs and Transitions

labelings of arcs:



labeling

labeling of a transition:



Arcs and transitions can be labeled with *expressions*. Next to elements of the real world, which have occurred in markings before, functions (for instance, a subtraction) and variables (for instance, x and y) can occur in such expressions. These expressions have a central property: if all variables in an expression are replaced by elements, it becomes possible to evaluate the expression in order to obtain yet another element. It is convenient to write the labeling of an arc (p, t) or (t, p) as

$$\overline{pt} \text{ or } \overline{tp} \quad (1)$$

respectively. Statement (1) describes the tokens that “flow through the arc” at the occurrence of t .

The variables in these expressions are *parameters* describing different instances (“modes”) of a transition. Such a transition can only occur if its labeling evaluates to the logical value “true”. The rest of this chapter describes this correlation in more detail.

2.3 The Data Structure for Petri Nets: Multisets

In a Petri net, the tokens of a place often represent objects that we usually do not want to distinguish. For instance, we are only interested in the number of coins in the cash box (although we could, for instance, distinguish them by their date of coining).

In general, examples of different *kinds of tokens* are mixed in a place, e.g., rectangular and round cookie packets in the storage H. They form a *multiset* a , formally a mapping

$$a : U \rightarrow \mathbb{N}$$

that maps every kind u of a *universe* U to the number of its occurrences in a .

We always assume a “sufficiently large” universe U that contains all examined kinds of tokens. We write the set of all multisets over U as

$$\mathcal{M}(U) \text{ or } \mathcal{M} \text{ for short}$$

if the context unambiguously identifies the universe U .

The universe U can contain an infinite number of elements, for instance, all natural numbers. A multiset a over U can map the value $a(u) = 0$ to almost all $u \in U$. That means that u does not occur in a . Thus, a is *finite* if

$$a(u) \neq 0 \text{ for only a finite number of } u \in U.$$

We write a finite multiset a with its multiple elements in square brackets $[\dots]$. Consequently, the *empty* multiset is denoted by $[\]$:

$$[\](u) = 0 \text{ for each } u \in U.$$

Multisets $a, b \in \mathcal{M}$ can be added: for each $u \in U$, let

$$(a + b)(u) =_{\text{def}} a(u) + b(u).$$

They can be compared:

$$a \leq b \text{ iff for each } u \in U : a(u) \leq b(u),$$

and b can be subtracted from a if $b \leq a$:

$$(a - b)(u) =_{\text{def}} a(u) - b(u).$$

With these notations, we describe dynamic behavior.

The tokens of a place p usually belong to a *type*, that is, to a (small) subset of the universe U . In Fig. 2.1, for instance, only coins lie in the places A and F, only black tokens in D and G, only cookie packets in H and C and only numbers in E. If a place p only holds tokens of type τ , the type τ is assigned to the place p .

multiset



multiset a of H:

$$\begin{aligned} a(\text{rect}) &= 3 \\ a(\text{round}) &= 2 \\ a(u) &= 0, \quad \text{for any other } u \end{aligned}$$

set $\mathcal{M}(U)$ of all multisets

the universe of the cookie vending machine:

$$\text{rect}, \text{round}, \text{coin}, 0, 1, 2, 3, 4, 5, 6, 7, \dots$$

finite multiset

finite multiset a :

$$[\text{rect}, \text{rect}, \text{rect}, \text{round}, \text{round}]$$

$$a(\text{rect}) = 3, a(\text{round}) = 2$$

empty multiset $[\]$

sum of multisets

order on multisets

subtraction on multisets

arithmetic operations on multisets:

$$[\text{rect}, \text{round}] + [\text{rect}] = [\text{rect}, \text{rect}, \text{round}]$$

$$[\text{rect}, \text{round}] \leq [\text{rect}, \text{rect}, \text{round}]$$

$$a \leq a$$

$$[\text{rect}, \text{round}, \text{round}] - [\text{round}] = [\text{rect}, \text{round}]$$

$$a - a = [\]$$

2.4 Markings as Multisets

marking

Initial marking M_0 of Fig. 2.1:

$$M_0(H) = [\text{cookie}, \text{cookie}, \text{cookie}, \text{cookie}, \text{cookie}, \text{cookie}, \text{cookie}]$$

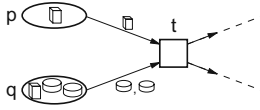
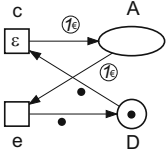
$$M_0(A) = M_0(B) = M_0(C) = []$$

$$M_0(D) = M_0(G) = [\bullet]$$

$$M_0(E) = [7]$$

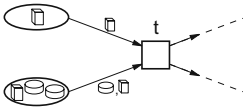
$$M : P \rightarrow \mathcal{M}.$$

That means that M maps every place p to a multiset $M(p)$. As explained in Sect. 2.2, a marking M describes a *state* of the modeled system. Given the significance of a system's initial state, the initial marking (usually denoted M_0) is often drawn into the respective net structure.



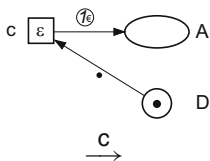
$$\overline{pt} = [\text{cookie}]$$

$$\overline{qt} = [\text{cookie}, \text{cookie}]$$



t is enabled

the first step of the cookie vending machine:



2.5 Steps with Constant Arc Labelings

Let us now examine the special case in which the arcs around a transition t are labeled with individual elements of a universe. This applies to the arcs around the transitions c and e in Fig. 2.1. In general, however, an arc is labeled with more than one element. Formally, this is a multiset, whose brackets [and] are not written in order to save space. Thus, with the notation of (1), for each arc (p, t) or (t, p) the following holds:

$$\overline{pt} \in \mathcal{M} \quad \text{and} \quad \overline{tp} \in \mathcal{M}.$$

We technically expand this notation for all places p and transitions t by

$$\overline{pt} = [] \quad \text{and} \quad \overline{tp} = []$$

if no arc (p, t) or (t, p) exists, respectively.

A transition t can *occur* in a marking M if the related preconditions are met, that is if M *enables* the transition t .

As in many other system models, we separate the *enabling* of t from the effect of the *occurrence* of t . Whether a marking M enables a transition t depends on the labelings of the arcs ending in t . M *enables* t if and only if

$$M(p) \geq \overline{pt}$$

for each arc (p, t) . Thus, the initial marking of the cookie vending machine only enables the transition c .

If a marking M enables a transition t , it results in the *step*

$$M \xrightarrow{t} M',$$

in which the marking M' of each place p is defined as

$$M'(p) =_{\text{def}} M(p) - \overline{pt} + \overline{tp}.$$

2.6 Steps with Variable Arc Labelings

An arc or a transition can be labeled with an expression a that contains variables. By assigning values to the variables in a , the expression a can be evaluated. If a is written onto an arc, the result is a multiset. If a is written into a transition, the result is either “*true*” or “*false*”. In order to calculate these values, the labelings of all arcs that end or start at a transition (the arcs *around* t) have to be taken into account simultaneously.

Put a little more technically: Let x_1, \dots, x_n be the variables of the arc labelings around a transition t . Let u_1, \dots, u_n be elements of the universe. Then

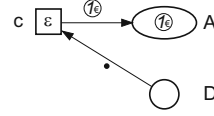
$$\beta : (x_1 = u_1, x_2 = u_2, \dots, x_n = u_n)$$

is a *mode* of t . In Fig. 2.1, for instance, the variables y and z occur in the arc labelings around the transition b . Thus, $\beta_1 : (y = \boxed{}, z = \bigcirc)$ is a mode of b . The transition b has three additional modes: $\beta_2 : (y = \bigcirc, z = \boxed{})$, $\beta_3 : (y = z = \bigcirc)$ and $\beta_4 : (y = z = \boxed{})$. A mode β of a transition t creates for each arc (p, t) or (t, p) a multiset $\beta(p, t)$ or $\beta(t, p)$, respectively. Thus, in Fig. 2.1, $\beta_1(H, b) = \beta_2(H, b) = [\boxed{}, \bigcirc]$. Another example is $\beta : (x = 7)$, a mode of the transition t in Fig. 2.1, and it holds: $\beta(a, E) = [7 - 2] = [5]$.

If \overline{pt} does not contain any variables, then obviously

$$\beta(p, t) = \overline{pt}.$$

A transition t can itself have a labeling that contains variables. An example is the labeling $x \geq 2$ of transition a in Fig. 2.1. For such a labeling i , a mode β of t creates a logical value, $\beta(i)$. For instance, for the labeling $x \geq 2$ of a , the mode $\beta_1 : (x = 7)$ creates the logical value $\beta_1(x \geq 2) = [7 \geq 2] = \text{true}$.



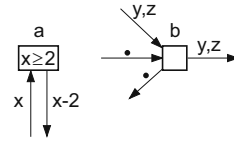
formally:

$$\begin{aligned} M_0 &\xrightarrow{c} M_1 \\ M_1(D) &= [\], M_1(A) = [\textcircled{\bullet}] \\ M_1(p) &= M_0(p), \\ &\text{for any other place } p \end{aligned}$$

expression

transition condition

arc labeling

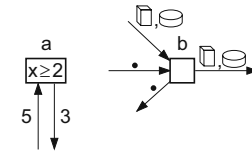


mode of a transition

mode β of the transitions a and b :

$$\beta(x) = 5, \beta(y) = \boxed{}, \beta(z) = \bigcirc$$

results in the constant arc labelings



Thus, a mode β of t creates multisets at the arcs around t . A step of t in the mode β is then defined as described in the previous section. Additionally, the labeling a of t has to evaluate to $\beta(a) = \text{true}$. For a step from M to M' via t in the mode β , we write

$$M \xrightarrow{t, \beta} M'.$$

The symbol β for the mode is often omitted and we write, for instance

$$x = 5 \text{ instead of } \beta : (x = 5).$$

M enables t in mode β

Put in a formal context, a marking M enables a transition t in the mode β of t if for each arc in the form (p, t) :

$$M(p) \geq \beta(p, t)$$

and for the labeling i of t :

$$\beta(i) = \text{true}.$$

This then results in the step $M \xrightarrow{t, \beta} M'$, in which M' for each place p is defined by

$$M'(p) = M(p) - \beta(p, t) + \beta(t, p).$$

Again, let $\beta(p, t) = []$ and $\beta(t, p) = []$ if no arc (p, t) or (t, p) exists in N , respectively.

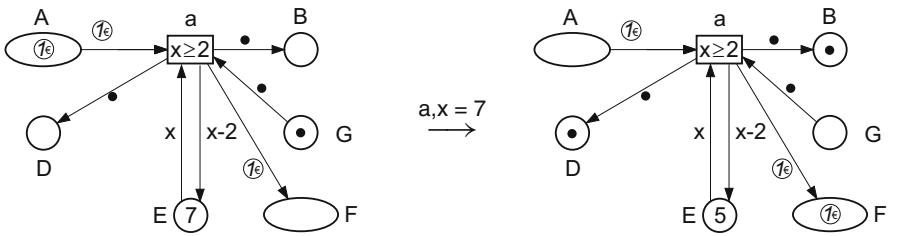
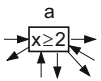


Figure 2.3 Step $M_1 \xrightarrow{a, x=7} M_2$

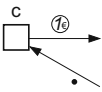


for $\beta(x) = 7$:
condition $x \geq 2$ is met!

Consider the cookie vending machine in Fig. 2.1: After the step $M_0 \xrightarrow{c} M_1$, the marking M_1 enables the transition a in the mode $x = 7$, and in no other mode. Figure 2.3 shows the effect that the step

$$M_1 \xrightarrow{a, x=7} M_2$$

has on the surroundings of the transition a. The marking M_2 , which is then reached, enables the transition b, because now values can be assigned to the variables y and z. Every assignment of \square or \bigcirc to these variables enables b. Thus, there exists a selection of four modes and hence four steps in M_2 , as outlined in Fig. 2.4.



c has only one mode

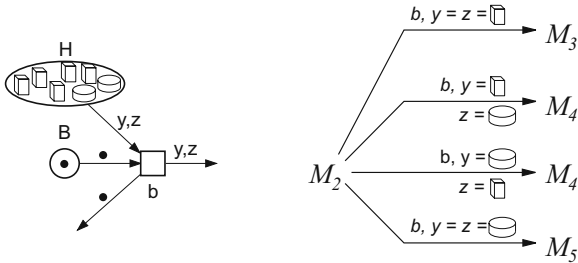


Figure 2.4 M_2 enables b in four modes

2.7 System Nets

We have now assembled the principal notations that enable us to describe a discrete, dynamic system, as for instance a cookie vending machine. According to the principles in Section 2.2, we use an appropriately labeled, finite net structure, N , to do this. A central term is that of a *marking of N* , that is, a distribution of tokens (multisets) across the places of N . Typically, the *initial marking* of N is denoted by M_0 and is explicitly drawn into N . M_0 describes the initial state of the modeled system. A transition t can be labeled with a *condition* and the arcs around t with *expressions*. These labelings show the various situations (*modes*) in which t is enabled, and the respective effects at the occurrence of t . Lastly, every transition is either *hot* or *cold*, where cold transitions are indicated by “ ϵ ”.

cold transition

A net structure together with an initial marking, transition conditions, arc labelings and cold transitions form a *system net*.

system net

System nets are used to model real, discretely changeable systems. Each place of a system net models a state component of the system and each currently existing token in a place models a currently given, but changeable, characteristic of that component. Each transition of a system net represents an action of the system. The occurrence of a transition describes the occurrence

of the respective action. If, in doing so, a token reaches or leaves a place, the action respectively creates or terminates the corresponding characteristic of the state component.

2.8 Marking Graph

For a system net N and an initial marking M_0 , a marking M of N is *reachable* if there exists a sequence of steps

$$M_0 \xrightarrow{t_1, \beta_1} M_1 \xrightarrow{t_2, \beta_2} \dots \xrightarrow{t_n, \beta_n} M_n$$

with $M_n = M$. In general, infinitely many markings of N are reachable. The reachable markings and steps of a system net N can be compiled into the *marking graph* of N . Its nodes are

reachable marking

marking graph

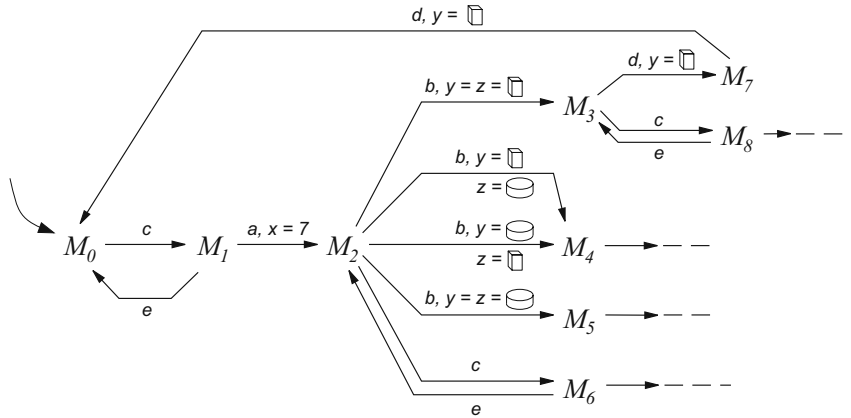


Figure 2.5 Initial part of the marking graph for the system net in Fig. 2.1

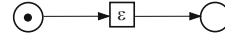
the reachable markings, its edges the steps between the reachable markings of N . The initial marking M_0 of N is specifically highlighted. The marking graph is also often called the *reachability graph*. Figure 2.5 shows an initial part of the marking graph for the system net in Fig. 2.1. The complete marking graph has approximately 100 nodes. In contrast to the system net in Fig. 2.1, it would be extremely laborious and counterintuitive to use the marking graph as a model for the cookie vending machine. In principle, the marking graph of a system net is a suitable starting point for its (automated) analysis, as long as only a finite number of markings are reachable.

2.9 Final Markings

A system has reached a final state if it can remain in this state forever. The marking of the system net in Fig. 1.9 models such a state, in contrast to Fig. 1.10 and Fig. 1.11. A final state of a system corresponds to a *final marking*. In such a marking, no hot transitions are enabled. For instance, in the system net in Fig. 2.1, the initial marking is, at the same time, also a final marking (it only enables the cold transition c).

final marking

a final marking:



not a final marking:



Exercises

1. The system net in Fig. 2.6 expands the cookie vending machine in Fig. 2.1 by a transition f . In your own words, describe the effect and the function of f inside the cookie vending machine.

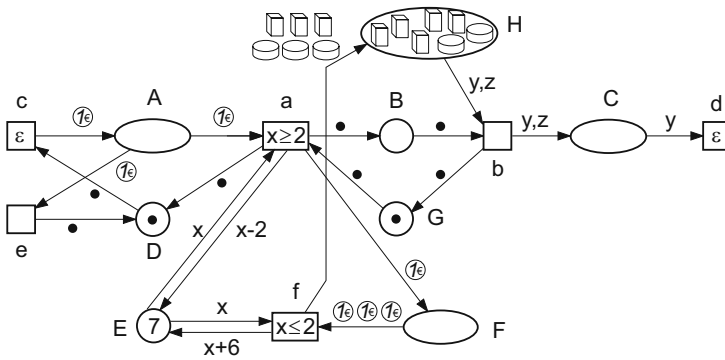


Figure 2.6 Expansion of the cookie vending machine in Fig. 2.1 by a transition f

2. Which of the markings M_0, \dots, M_8 in the marking graph in Fig. 2.5 are final markings?

Further Reading

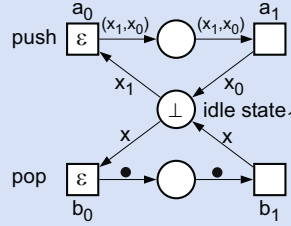
In the first two chapters of this book, we break with tradition in introducing the field of Petri nets. Usually, one begins with the technically simple case of a single kind of “black” token, a case we will not cover until the next chapter. Instead we have immediately introduced “individual” tokens, because they are intuitively more comprehensible, more realistic and more accurate. The price for this is a complex step rule. For such nets, the literature gives many more, ultimately equally expressive, representations. Widely used is a version of Petri nets called *colored nets* [38]. They emphasize the semantics of functions over multisets. Girault and Valk [29] also use such nets.

As is often traditional in mathematics, we do not always distinguish objects and functions from their symbolic representations in expressions, equations, etc. Our distinction between hot and cold transitions is found only sporadically in the literature on Petri nets. Damm and Harel used it in *Live Sequence Charts* [35] as a very apt way of expressing system specifications.

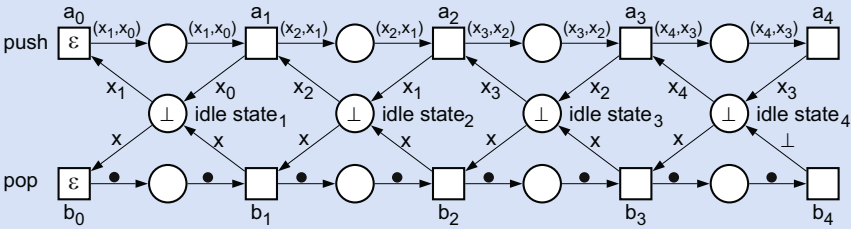
In the historical development, Genrich and Lautenbach [28] introduced nets with *individual* tokens as *predicate/transition nets* and in doing so emphasized the connection with logic.

A Universal, Expandable Architecture

In his dissertation, Petri designed a universal computer architecture that can be expanded an arbitrary number of times. We will explain this idea using the example of a finite, but infinitely expandable stack. It consists of a sequence $A_0 \dots A_n$ of *modules*, where each module A_i ($i = 0, \dots, n$) has an *idle state* that stores either a value or – e.g., initially – a “dummy” \perp . The net



shows the module A_0 with its interface to the environment: The transition push accepts a value from the environment via the variable x_0 , which may hold any value. The variable x_1 holds the previously stored value, which is passed on to the module A_1 via the transition a_1 . The transition pop extracts the value stored in idle state. Via b_1 the module then receives the value stored in A_1 . The net



combines four modules to form a stack. Each module A_i behaves according to the pattern described for A_0 . Each occurrence of push or pop triggers a wave that moves from left to right through the stack. It ends in A_4 by popping out the previously stored value or pushing in a “dummy” \perp respectively. The transitions a_4 and b_4 are the extension points where another module A_5 can be attached.

Understanding Petri Nets

Modeling Techniques, Analysis Methods, Case Studies

Reisig, W.

2013, XXVII, 230 p. 145 illus., Hardcover

ISBN: 978-3-642-33277-7