1 Petri nets

A Petri net $N = (P, T, F)$ consists of a finite set P of *places*, a finite set T of *transitions* such that P and T are disjoint, and a *flow relation* $F \subseteq (P \times T) \cup (T \times P)$.

The set $t^\bullet = \{p \mid (t, p) \in F\}$ defines all output places of a transition t. We refer to ${}^\bullet t$ as the *preset of* t and to $t^\bullet$ as the *postset of* t.

A *marking* of a Petri net (P, T, F) is a function $m : P \to \mathbb{N}$, assigning to each place $p \in P$ the number $m(p)$ of tokens at this place.

A Petri net system (P, T, F, m_0) consists of a Petri net (P, T, F) and a distinguished marking m_0, the *initial marking*.

Exercise 1.1 Draw the Petri net defined by $P = \{p_1, p_2\}$, $T = \{t_1, t_2\}$, and $F = \{(p_1, t_1), (t_1, p_2), (p_2, t_1), (p_2, t_2), (t_2, p_2), (t_2, p_1)\}$.

Exercise 1.2 Draw the Petri net system (P, T, F, m_0), which is defined as $P = \{p_1, \dots, p_7\}$, $T = \{t_1, \dots, t_6\}$, $F = \{(p_1, t_1), (p_2, t_2), (p_3, t_3), (p_4, t_1), (p_4, t_4), (p_5, t_5), (p_6, t_6), (p_7, t_4), (t_1, p_2), (t_2, p_3), (t_3, p_1), (t_3, p_4), (t_4, p_5), (t_5, p_6), (t_6, p_4), (t_6, p_7)\}$, and $m_0 = [p_1, p_4, p_7]$.

Exercise 1.3 Consider the Petri net in Figure 1.

1. Define the net formally as a triple (P, T, F).
2. List for each transition what its preset and postset are.

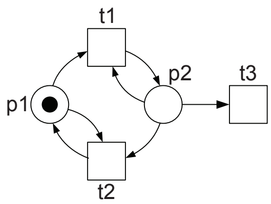

Figure 1: A Petri net.

Exercise 1.4 Consider the Petri net in Figure 1 and assume a marking m where every place contains exactly one token.

1. Which transitions are enabled at m?
2. What is the result of the firing of transition t_1 in m?
3. What is the result of the firing of transition t_2 in m?
4. What is the result of the firing of transition t_3 in m?
5. What are the reachable markings from m, and which of these markings are terminal markings?

Exercise 1.5 Consider the Petri net system in Figure 2.

1. Formalize this net as a four-tuple (P, T, F, m_0).
2. Give the preset and the postset of each transition.
3. Which transitions are enabled at m_0?
4. Give all reachable markings.
5. What are the reachable terminal markings?

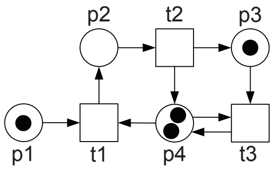

Figure 2: A Petri net system.

6. Is there a reachable marking where we have a nondeterministic choice?

7. Does the number of reachable markings increase or decrease if we remove (1) place p_1 and its adjacent arcs and (2) place p_3 and its adjacent arcs?

Exercise 1.6 Consider the Petri net system in Figure 3.

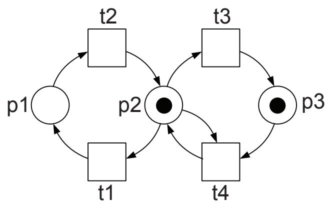

Figure 3: A Petri net system.

1. Formalize this net as a four-tuple (P, T, F, m_0).

2. Give the preset and the postset of each transition.

3. Which transitions are enabled at m_0?

4. Give all reachable markings.

5. What are the reachable terminal markings?

6. Is there a reachable marking where we have a nondeterministic choice?

7. Does the number of reachable markings increase or decrease if we remove place p_1 and its adjacent arcs?

Exercise 1.7 The firing of a transition t at marking m yields a successor marking m'. Function $m'(p)$ in the definition of firing defines the effect of transition t on a place p. It is any one of $m(p)+1$, $m(p)-1$, or $m(p)$. Explain when each of the three effects occurs and formalize this by completing the following equation:

$$m'(p) = \begin{cases} m(p)-1, & \text{if } \ldots, \\ m(p)+1, & \text{if } \ldots, \\ m(p), & \text{if } \ldots. \end{cases}$$

Exercise 1.8 Represent the Petri net system in Figure 4 as a transition system assuming that

1. Each place cannot contain more than one token in any marking;

2. Each place may contain any natural number of tokens in any marking.

Hint: Describe a marking of the net as a triple (x, y, z) with x specifies the number of tokens in place *free*, y in place *busy*, and z in place *docu*.

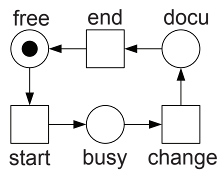

Figure 4: A Petri net.

Exercise 1.9 A plant produces an item to order by a grinding step (event "g"), followed by a milling step (event "m"). After the milling step, the item is tested. If the test is positive (event "positive"), the item is sent to the customer (event "send") and the production of the next item can start; otherwise (event "negative"), the item is discarded and the production starts again with a grinding step. Items can be in any of the states "wait", "grinded", "milled", "to_be_sent", "sent", and "discarded". Suppose that there are initially two items in state "wait". Model this business process as a Petri net system.

Exercise 1.10 Model the following treatment of a patient at a dentist as a Petri net system. If we look at the patient separately, then there are five states: the patient is at home (state "p_home"), the patient is sitting in the waiting room (state "p_wait"), the patient is treated (state "p_treat"), the treatment is finished (state "p_done"), and the patient has left the practice (state "p_left"). There are four events: the patient enters the practice ("enter"); the treatment starts ("start"); the treatment is documented ("docu"); and the treatment ended ("end"). The dentist can be in three states: "d_free", "d_busy", and "d_docu". There are three respective events: "start", "docu", and "end". The nurse can be either in state "n_free" or in state "n_busy". There are two events where the nurse is involved: "start" and "docu". The nurse starts being busy after event "start" and becomes free again after event "docu", namely when the patient is not treated anymore. Finally, there is also a secretary involved. Either she sits at the reception (state "s_reception") or she helps the dentist documenting the treatment and writing a prescription (state "s_docu"). For the documentation, there are two events relevant: "docu" and "end". Furthermore, when a patient enters the practice, the secretary is taking care of the reception. Suppose that initially there are two patients at home, the dentist and the nurse are in state "free", and the secretary is sitting at the reception. Hint: Model first the patient and add then the dentist, the secretary, and the nurse.

Exercise 1.11 From a bridge, frogs jump into a stream, nondeterministically choose one of the two banks to swim to, and then hop to the bridge to start over again. A lovely girl picks up every third frog from the stream, kisses the frog, and puts the frog back on the bridge. Model this fairy tale as a Petri net. Suppose that three frogs are initially on the bridge.

Exercise 1.12 Consider the handling of insurance claims at Sunny Side Corp., Australia. Sunny Side distinguishes simple claims and complex claims. The type of the claim is determined in the first step.

For simple claims, Sunny Side carries out two steps independently: it checks the insurance policy of the insured party for validity and retrieves the statement of a local authority. When both results are available, in a next step Sunny Side checks the statement against the policy. If the result is positive, Sunny Side makes a payment to the insured party; if the result is negative, Sunny Side sends a rejection letter.

For complex claims, Sunny Side carries out three steps independently: it checks the insurance policy of the insured party for validity, retrieves the statement of a local authority, and asks for two witness statements. The business process can only proceed if both witness statements are available. Again, when both results are available, in a next step Sunny Side checks the statements of the local authority and witnesses against the policy. If the result is positive, Sunny Side makes a payment to the insured party; if the result is negative, Sunny Side sends a rejection letter.

1. Model this business process as a Petri net.

Exercise 1.13 Consider the following business process for handling traffic offenses. Every offense is registered after arrival. After registration, procedures "judge the traffic offense" and "investigate the history" are started concurrently. In procedure "judge the traffic offense", the traffic offense is classified as either "severe" or "normal". Severe traffic offenses are then temporary judged, and, in a second step, a final judgment is delivered. Normal traffic offenses are judged in one step. Procedure "investigate the history" contains two steps that can be completed in arbitrary order: collect information about earlier traffic offenses and collect information about other offenses committed by the offender. The fine is determined after both procedures are finalized. If the traffic offense is not fined, it will be archived right away; otherwise, a transfer form is sent to the offender and subsequently the traffic offense will be archived.

1. Model this business process as a Petri net.

Exercise 1.14 Consider the process of a construction company to build houses according to orders from customers. The process depends on whether a completely new house is built, or whether there is still an old house at the building site.

If a completely new house is built, the company builds a new foundation, wooden frames for the walls, and wooden frames for the roofs independently of each other at the same time. As soon as the foundation and the frames for the walls are completed, the walls are put up. When the frames for the roof are completed and the walls have been put up, the roof is installed. Finally, the complete house is handed over to the customer.

If there is still an old house at the building site, first the roof and walls of the house are demolished together. Next, new wooden frames for the walls and for the roof are built independently of each other at the same time. As soon as the frames for the walls are completed, the walls are put up. Once the walls are put up and the frames for the roof are finished, an engineer checks the static stability of the wall frames and the roof frames. If both are ok, then the roof is installed. Otherwise, wall frames and roof frames are first disassembled together, and then both are built, put up, and checked again as before. Once completed, the house is handed over to the customer.

Solutions

Solution 1.1

> ***This solution is just an example. There are many possible solutions to any given modeling excercise.***

Figure 5 depicts the Petri net.

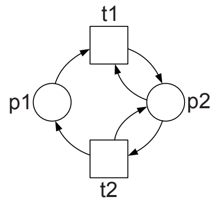

Figure 5: The Petri net of Exercise 1.1.

Solution 1.2

> ***This solution is just an example. There are many possible solutions to any given modeling excercise.***

Figure 6 depicts the Petri net.

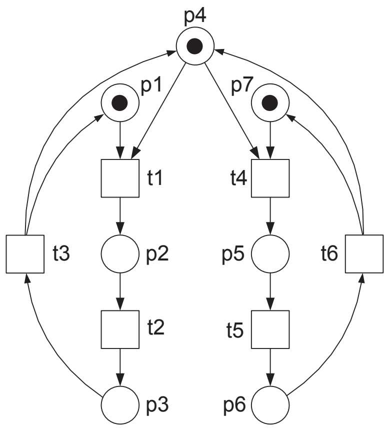

Figure 6: The Petri net of Exercise 1.2.

Solution 1.3

1. We obtain $P = \{p1, p2\}$, $T = \{t1, t2, t3\}$, and
   $F = \{(p1, t1), (t1, p2), (p2, t1), (p2, t3), (p2, t2), (t2, p1), (p1, t2)\}$.

2. We obtain the following presets and postsets: ${}^\bullet t1 = \{p1, p2\}$, $t1^\bullet = \{p2\}$, ${}^\bullet t2 = \{p1, p2\}$, $t2^\bullet = \{p1\}$, ${}^\bullet t3 = \{p2\}$, $t3^\bullet = \emptyset$.

Solution 1.4

1. All three transitions are enabled. If more than one transition is enabled, it is not clear which of these transitions will fire. A nondeterministic choice needs to be made.

2. If transition *t1* fires, then place *p1* is empty and place *p2* still contains one token.

3. If transition *t2* fires, then place *p2* is empty and place *p1* contains one token.

4. If transition *t3* fires, then place *p2* is empty and place *p1* contains one token.

5. The following markings are reachable: $[p1, p2]$ (i.e., the initial marking), $[p2]$ (i.e., the marking after the firing of transition $t1$), $[p1]$ (i.e., the marking after the firing of transition $t2$), $[\,]$ (i.e., the marking after the firing of transition $t3$ in marking $[p2]$), and $[p1]$ (i.e., the marking after the firing of transition $t3$ in the initial marking $[p1, p2]$). Markings $[p1]$ and $[\,]$ are the reachable terminal markings.

Solution 1.5

1. We obtain $P = \{p1, p2, p3, p4\}$, $T = \{t1, t2, t3\}$,
$F = \{(p1, t1), (t1, p2), (p2, t2), (t2, p4), (t2, p3), (p3, t3), (t3, p4), (p4, t3), (p4, t1)\}$, and $m_0 = [p1, p3, 2 \cdot p4]$.

2. We obtain ${}^\bullet t1 = \{p1, p4\}$, $t1^\bullet = \{p2\}$, ${}^\bullet t2 = \{p2\}$, $t2^\bullet = \{p3, p4\}$, ${}^\bullet t3 = \{p3, p4\}$, $t3^\bullet = \{p4\}$

3. Transitions $t1$ and $t3$ are enabled at m_0, because each of their respective input places contains at least one token. Transition $t2$ is not enabled at m_0, because its input place $p2$ does not contain a token.

4. The initial marking $m_0 = [p1, p3, 2 \cdot p4]$. The firing of transition $t3$ yields marking $[p1, 2 \cdot p4]$. Afterward, transitions $t1$, $t2$, and $t3$ can subsequently fire, yielding markings $[p2, p4]$, $[p3, 2 \cdot p4]$, and $[2 \cdot p4]$. If transition $t1$ fires in m_0, we reach marking $[p2, p3, p4]$. In this marking, transition $t3$ can fire, yielding marking $[p2, p4]$ (the reachable markings of this marking have been already given). Otherwise, if transition $t2$ fires in marking $[p2, p3, p4]$, the net reaches marking $[2 \cdot p3, 2 \cdot p4]$ and by firing transition $t3$ twice, we reach markings $[p3, 2 \cdot p4]$ and $[2 \cdot p4]$.

5. The only reachable terminal marking is $[2 \cdot p4]$.

6. At marking $[p1, p3, 2 \cdot p4]$, transitions $t1$ and $t3$ are enabled; and at marking $[p2, p3, p4]$, transitions $t2$ and $t3$ are enabled.

7. If we remove place $p1$, then we can have an infinite firing sequence: $\langle t1, t2, t1, \dots \rangle$. As each firing of transition $t2$ produces a token in place $p3$, we can reach infinitely many different markings, meaning, the number of markings increases. In contrast, removing place $p3$ reduces the number of reachable markings. The intuition is that the firing of transition $t3$ does not change the marking, whereas this was the case in the presence of place $p3$.

Solution 1.6

1. We obtain $P = \{p1, p2, p3\}$, $T = \{t1, t2, t3, t4\}$,
$F = \{(p1, t2), (t2, p2), (p2, t1), (t1, p1), (p2, t3), (t3, p3), (p3, t4), (t4, p2), (p2, t4)\}$, and $m_0 = [p2, p3]$.

2. We obtain ${}^\bullet t1 = \{p2\}$, $t1^\bullet = \{p1\}$, ${}^\bullet t2 = \{p1\}$, $t2^\bullet = \{p2\}$, ${}^\bullet t3 = \{p2\}$, $t3^\bullet = \{p3\}$, ${}^\bullet t4 = \{p2, p3\}$, $t4^\bullet = \{p2\}$.

3. Transitions $t1$, $t3$, and $t4$ are enabled at m_0, because each of their respective input places contains one token. In contrast, transition $t2$ is not enabled at m_0, because its input place $p1$ does not contain a token.

4. The initial marking is $[p2, p3]$. The firing of transition $t1$ yields marking $[p1, p3]$. By firing transition $t2$, m_0 is reached again. The firing of transition $t3$ in m_0 yields marking $[2 \cdot p3]$. The firing of transition $t4$ in m_0 yields marking $[p2]$. If we continue with the firing of transition $t3$, we reach marking $[p3]$. Firing transitions $t1$ and $t2$ subsequently in marking $[p2]$ yields marking $[p1]$ and then marking $[p2]$ again.

5. There are two reachable terminal markings: $[p3]$ and $[2 \cdot p3]$.

6. At marking $[p2, p3]$, transitions $t1$, $t3$, and $t4$ are enabled; and at marking $[p2]$, transitions $t1$ and $t3$ are enabled.

7. If we remove place *p1*, the postset of transition *t2* is the empty set; that is, transition *t2* has no input places. As a consequence, transition *t2* is now in any reachable marking enabled (Note that the definition of an enabled transition "all input place of transition *t2* contain at least one token" is equivalent to "there is no input place of transition *t2* that contains less than one token"). So transition *t2* can subsequently fire infinitely often, thus producing an infinite number of tokens in place *p2*. Accordingly, infinitely many markings are reachable in the modified net.

Solution 1.7

If place p is an input place of transition t but no output place of t, then $m'(p) = m(p) - 1$. If place p is an output place of transition t but no input place of t, then $m'(p) = m(p) + 1$. The third case is less obvious. If place p is either no input and also no output place of transition t or input and output place of t, then we have $m'(p) = m(p)$. Clearly, if p is neither input nor output place of t, the number of tokens keep unchanged. Likewise, the number of tokens does not change if t produces a token in place p and consumes a token from p. Formally, we have

$$m'(p) = \begin{cases} m(p) - 1, & \text{if } p \in {}^{\bullet}t \setminus t^{\bullet}, \\ m(p) + 1, & \text{if } p \in t^{\bullet} \setminus {}^{\bullet}t, \\ m(p), & \text{if } (p \in {}^{\bullet}t \wedge p \in t^{\bullet}) \vee (p \notin {}^{\bullet}t \wedge p \notin t^{\bullet}) \end{cases}$$

As the third condition is fulfilled only if the first and second condition are not fulfilled, we can simplify the formalization as follows:

$$m'(p) = \begin{cases} m(p) - 1, & \text{if } p \in {}^{\bullet}t \setminus t^{\bullet}, \\ m(p) + 1, & \text{if } p \in t^{\bullet} \setminus {}^{\bullet}t, \\ m(p), & \text{otherwise.} \end{cases}$$

Solution 1.8

The transition system (S, TR, s_0) can be specified as follows. The set S of states is the set M of all markings of the Petri net system in Figure 4. The initial state is equal to the initial marking, and the transition relation TR contains all transitions that can occur from any of the states in S.

1. We obtain $S = \{(0,0,0), (1,0,0), (0,1,0), (0,0,1), (1,1,0), (1,0,1), (0,1,1), (1,1,1)\}$, $s_0 = (1,0,0)$, and $TR = \{((1,0,0),(0,1,0)), ((0,1,0),(0,0,1)), ((0,0,1),(1,0,0)), ((1,1,0),(1,0,1)), ((1,0,1),(0,1,1)), ((0,1,1)(1,1,0))\}$. Note that with the restriction of the number of tokens in a place to at most one, transitions like $((1,1,1),(0,2,1))$ are not possible.

2. We can formalize the states as $S = \{(x,y,z) \mid x,y,z \in \mathbb{N}\}$. The initial state s_0 is equal to the initial marking $m_0 = (1,0,0)$. The transition relation can be specified by the union of the following three sets:
$$\begin{aligned} TR = \; & \{((x+1,y,z),(x,y+1,z)) \mid x,y,z \in \mathbb{N}\} \\ & \cup \{((x,y+1,z),(x,y,z+1)) \mid x,y,z \in \mathbb{N}\} \\ & \cup \{((x,y,z+1),(x+1,y,z)) \mid x,y,z \in \mathbb{N}\}; \end{aligned}$$
The first, the second, and the third set contains all possible states that can be reached by firing transition *start*, *change*, and *end*, respectively.

Solution 1.9

This solution is just an example. There are many possible solutions to any given modeling excercise.

We model each event by a transition and each state by a place. Figure 7 depicts the corresponding Petri net system.

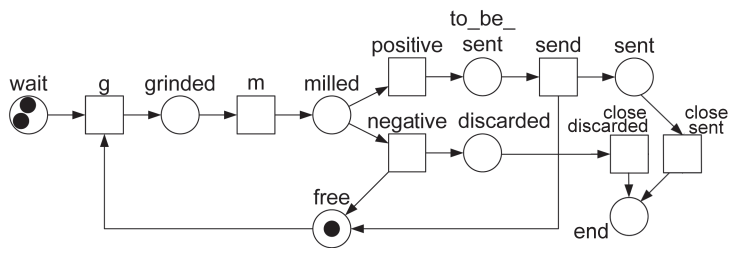

Figure 7: The Petri net system answering Exercise 1.9.

Solution 1.10

This solution is just an example. There are many possible solutions to any given modeling excercise.

We model each event as a transition and each state as a place. Figure 8 depicts the corresponding Petri net system. The modeling is straightforward. Observe that we model the step when a patient enters the practice and registers at the reception by consuming the token from place *s_reception* and by producing the token in place *s_reception*, because the secretary does not change her state.

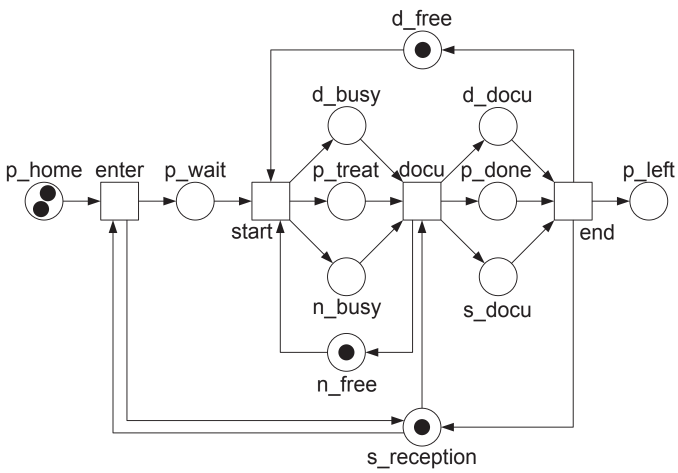

Figure 8: The Petri net answering Exercise 1.10.

Solution 1.11

This solution is just an example. There are many possible solutions to any given modeling excercise.

Figure 9 depicts the model of the fairy-tale. Let us first separately consider the states a frog can be in. A frog can be on the "bridge", in the "stream", on "bank1", or on "bank2". We model each of these states as a place. There are five events: the frog jumps from the bridge ("jump"), swims to one of the banks ("swim1" and "swim2"), or hops from the respective bank back on the bridge ("hop1" and "hop2"). We model each event as a transition. To model the girl, we need a place and a token in this place (state "girl") and an event "kiss" modeled as a transition *kiss*. To ensure that the girl picks up every third frog, we added transition *jump3* and place *stream3*. After transition *jump* fired twice, transition *jump3* is enabled. This is guaranteed by places *counter1* and *counter2*. When the girl kisses the frog and puts it back on the bridge (i.e., transition *kiss* fires), then two tokens are produced in place *counter2* and the next two frogs can jump using transition *jump*.

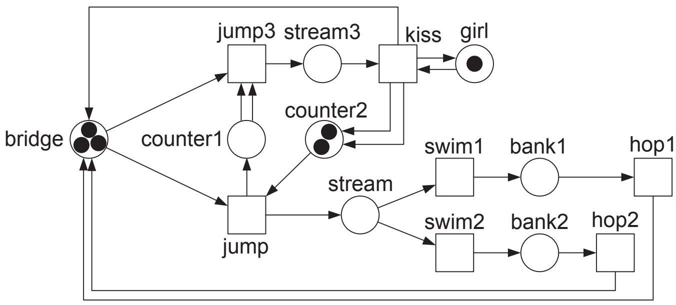

Figure 9: The Petri net of Exercise 1.11.

Figure 10 depicts another model of the fairy-tale. In contrast to Figure 9, in which every third frog is kissed in every run, the model in Figure 10 is less restrictive and only guarantees that every third frog is kissed in every infinite run, thereby assuming that the three transitions in the postset of place *bridge* fire with equal probability.

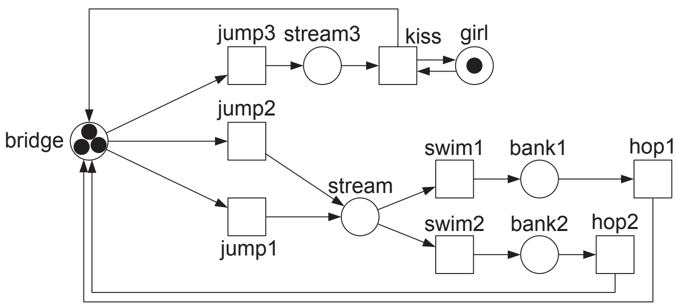

Figure 10: Another Petri net of Exercise 1.11.

Solution 1.12

This solution is just an example. There are many possible solutions to any given modeling excercise.

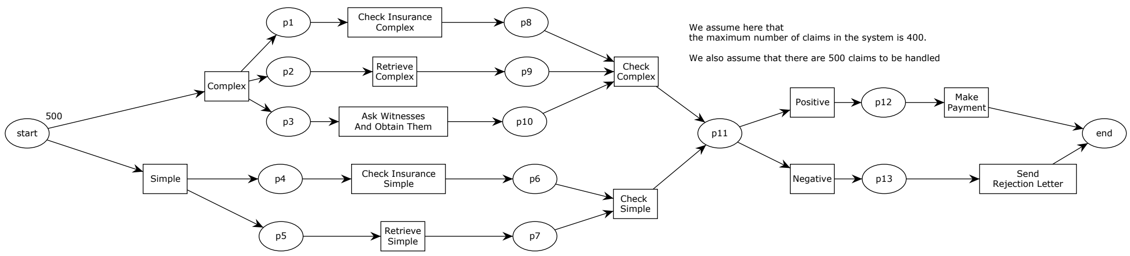

Figure 11: The Petri net of Exercise 1.12(1).

Figure 11 depicts the Petri net modeling this business process. Place *start* collects the incoming claims. Here, we assume, as a mere example, that 500 claims are ready for being dealt with. We model the decision whether a claim is simple or complex by a nondeterministic choice (transitions *Simple* and *Complex*). For simple claims, checking the insurance policy of the insured party for validity (transition *Check Insurance Simple*) and retrieving the statement of a local authority (transition *Retrieve Simple*) is performed concurrently. The result is checked against the policy (transition *Check Simple*). As this check may have two outcomes, positive and negative, two transitions (*Positive* and *Negative*) model the decision by a nondeterministic choice. In the complex case, three events are executed concurrently: checking the insurance policy of the insured party for validity (transition *Check Insurance Complex*), retrieving the statement of a local authority (transition *Retrieve Complex*), and asking for two witness statements (transition *Ask Witnesses and Retrieve Them*). Afterward, the results are checked against the policy (transition *Check Complex*). Afterwards, the flow continues with one of the same transitions as for the simple case (namely, *Positive* and *Negative*). If the positive branch is chosen, the process ends with making the payment (transitions *Make Payment*); otherwise, the rejection letter is sent.

Solution 1.13

This solution is just an example. There are many possible solutions to any given modeling excercise.

Figure 12 depicts the business process model. Place *p0* collects the arrived traffic offenses. The offenses are then registered (transition *register*). After registration, two procedures are performed concurrently. The first branch models the judgment of the traffic offense. A nondeterministic choice of transitions *normal* and *severe* models the decision whether the traffic offense is "normal" or "severe". If the traffic offense is normal, there is only a single judgment step (transition *judge*); otherwise, there are a temporary and a final judgment step (transitions *t_judge* and *f_judge*). The second branch investigates the history (transition *investigate_history*). Information about earlier traffic offenses (transition *earlier_offenses*) and about other offenses committed by the offender (transition *other_offenses*) are collected. Then the fine is determined (transition *determine*). We model the evaluation of the fine again as a nondeterministic choice (transitions *no_fine* and *fine*). In the case of a fine, first a letter is sent (transition *send*), and then the traffic offense is archived (transition *archive*). If no fine is determined, the traffic offense is immediately archived, and a token is produced in the input place of transition *archive*.

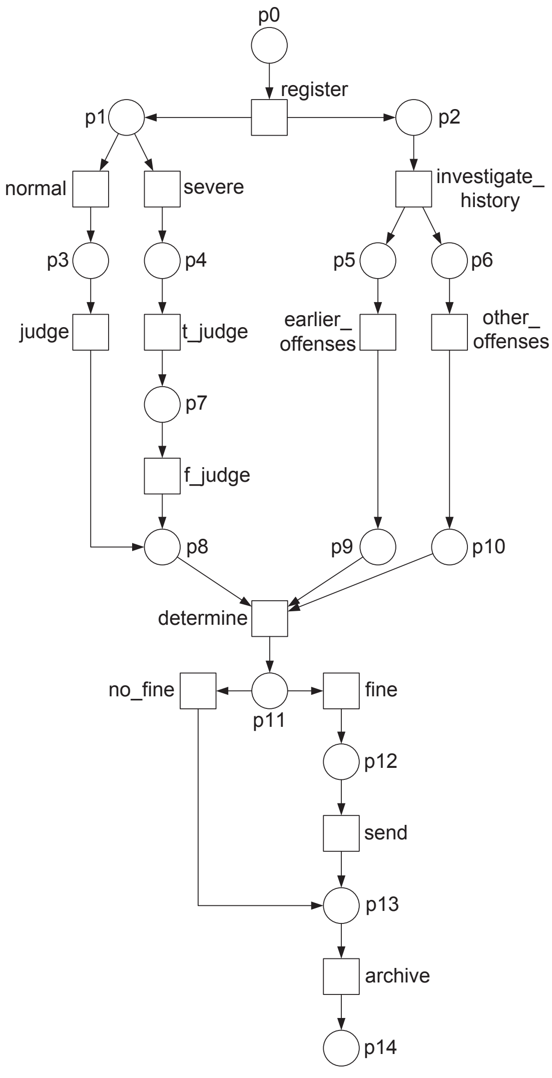

Figure 12: The Petri net of Exercise 1.13(1).

Solution 1.14

This solution is just an example. There are many possible solutions to any given modeling excercise.

Figure 13 depicts the Petri net. The red part is used to model the resource allocation and utilization.

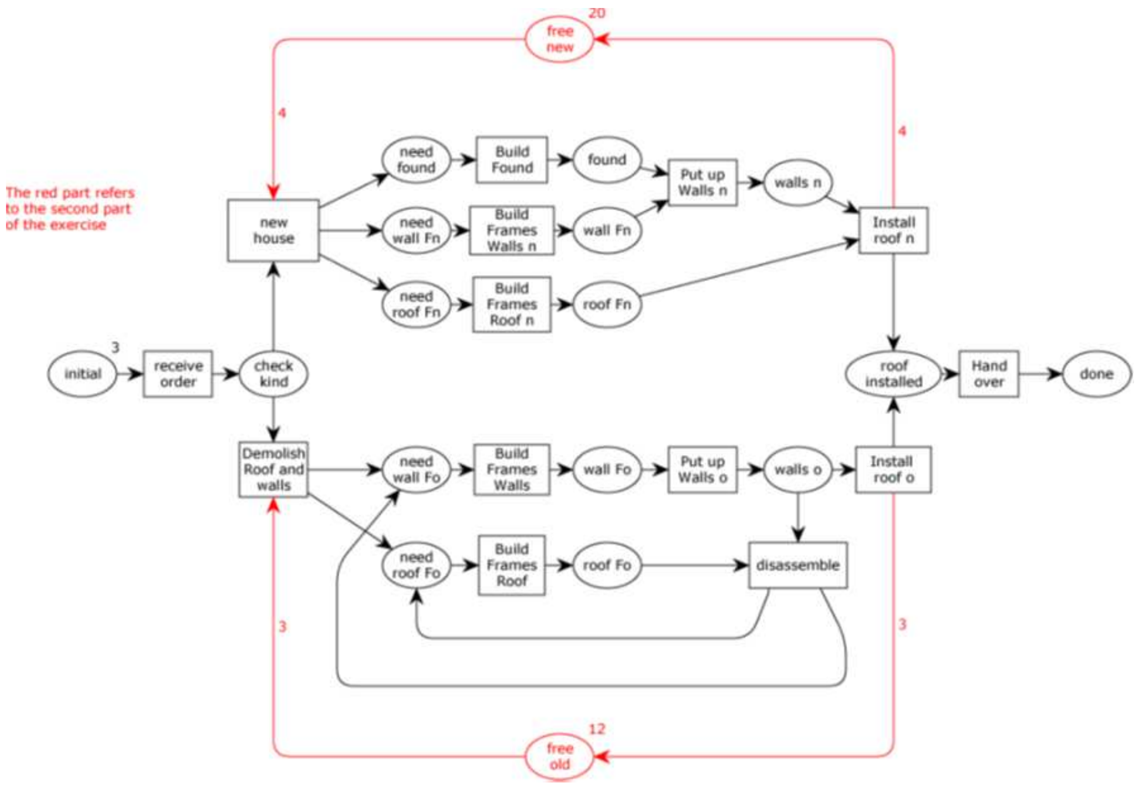

Figure 13: The Petri net of Exercise 1.14.