COPIED FROM:

Petri网通讯 第 2 期

中国计算机学会Petri网研究会 1989年11月

On Technical Safety and Security (continued)

C. A. Petri, C. Y. Yuan

October 20, 1989

A railway system is of course expected to be both *safe* and *live*. In other words, there should be no accidents caused by control logic and trains should be able to move one after another. This paper concentrates on the safety problem. In the previous paper the difference between train safety and token safety was discussed and based on this difference, (token) *security* as a system property was defined. What was done in that paper has set a good example for net application, since it is clear now that the difference between net concepts (e.g. tokens) and reality (e.g. trains) must be taken into account in net applications.

Our net models for trains should be live since otherwise it would be of no importance no matter how safe or secure they might be. We'll keep this demand of liveness in mind through this paper, but put no emphasis to it.

The purpose of this paper is:

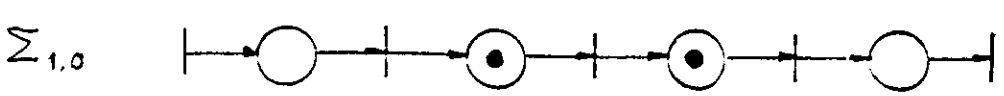
- to point out the essence of safety and security in more details; and
- to show the structure of *general* $n \times k$ secure transmission lines.

1 Evolution of Models

Several models were introduced in the previous paper (See Newsletter No 1).

1.1 One_Track Model

A model with one track and no control signal is as shown below:

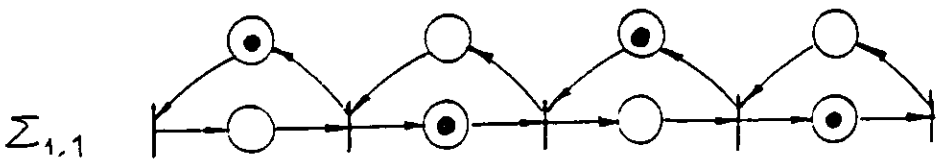


where trains are represented by tokens. It is clear that this model is not safe: the picture

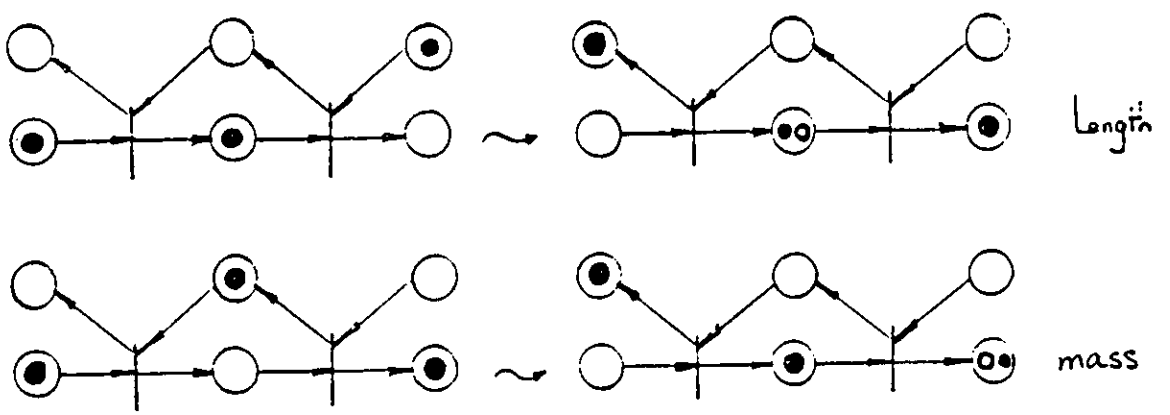
depicts a contact situation. To avoid contact, permit signals are employed to modify the model.

1.2 Model With One_Track and One_Signal_Line

In the following model, the signals are used to issue permit for trains to enter a track segment:



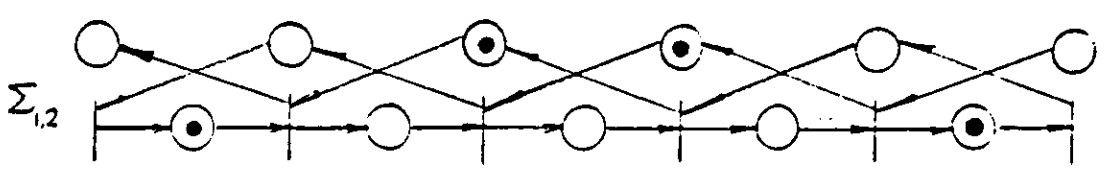
This model is safe as far as tokens are concerned. But trains are still not safe due to the nonzero length and nonzero mass nature of trains. The picture below illustrates why.



where represents unmodelled part of a train. This unmodelled part is either the tail of a train when train length is concerned, or the train head if the distance for a train to come to a full stop should be taken into account.
Trains have to be separated by empty segments in order to avoid such accidents. To this end, two or more signal lines are necessary.

1.3 Model With Two Signal_Lines

The following figure shows the model with two permit_signal lines:



Trains are now separated by at least one empty segment. It is usually the case in reality

that a railway segment (i.e. a section between two adjacent stations) is longer than the length of a train plus the distance needed for a train to come to a full stop. Theoretically, and generally speaking, we have the following formula:

$$n = \lfloor m \rfloor + 2$$

where

n is the number of segments reserved for a train (at least $n-1$ empty segments between two trains).
$m = Max\{l_\alpha/s + d_\beta/s \mid \alpha, \beta \text{ are trains}\}$
$\lfloor m \rfloor$ is the maximal integer no greater than m
$l_\alpha =$ length of train α
$d_\beta =$ distance needed for train β to come to a full stop
$s = Min\{length(\sigma) \mid \sigma \text{ is a segment}\}$

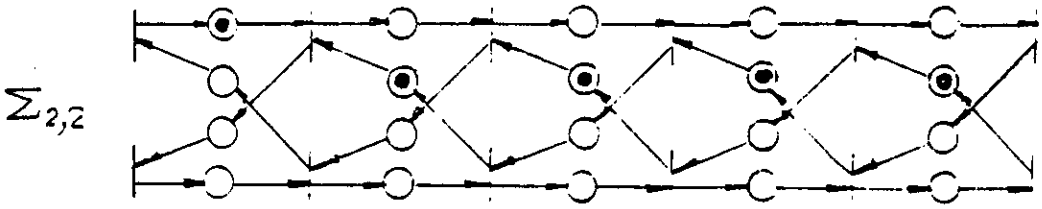
$\Sigma_{1,2}$ shows the case when $n = 2$. $n = 3$ was required in the previous paper so that one empty segment was reserved for train length and one empty segment for train mass. We are better off now since the above formula tells precisely how to add up the two factors together.

Note that m is in fact the maximal length of all trains plus the maximal braking distance for all trains.

As pointed out in the previous paper, the trains are still not safe in this model due to the nonzero length of signals. That is to say, signals should be separated as well, say by at least one segment on the signal line.

1.4 Model With Two Tracks and Two Signal Lines

By "two tracks" we mean one railway track and one signal line with the signals going in the direction the trains go. Signals on this line are not permit signals.



Similar to the way in which trains are separated from each other, two tracks (two lines in the train direction) leave at least one empty segment between two adjacent permit signals.

Assuming that the length of (permit) signals is less than the length of a segment on a signal line, we can now conclude that the tokens as well as the trains, when $n = 2$, are *safe* in $\Sigma_{2,2}$. With train safety, an informal concept in railway systems, as our goal in developing the net model, we have arrived at $\Sigma_{2,2}$. It has turned out that this model is not only *safe* in the formal sense in net theory, but also secure as defined in the previous paper. Now, *security* as a formal concept in net theory resembles the informal safety of trains in reality.

Let us recall the definition of security:

Definition

$$\text{Secure(S,T;F,C)}: \Longleftrightarrow$$
$$\bigwedge t \in T, \bigwedge c \in C:$$
$$\left.\begin{array}{lll} 1) & {}^\bullet t \subseteq c \rightarrow t^\bullet \cap c = \emptyset & \textit{no forward contact} \\ 2) & t^\bullet \subseteq c \rightarrow {}^\bullet t \cap c = \emptyset & \textit{no backword contact} \end{array}\right\} \text{"}safe\text{"}$$
$$\left.\begin{array}{lll} 3) & {}^\bullet t \cap c \neq \emptyset \rightarrow t^\bullet \cap c = \emptyset & \\ 4) & t^\bullet \cap c \neq \emptyset \rightarrow {}^\bullet t \cap c = \emptyset & \end{array}\right\} \textit{no transjunction};\ 3) \Longleftrightarrow 4)$$

What is new in the concept of security is "transjunction", short for "con*junction* of conditions across a *trans*ition".

2 Transjunction

Definition: A transition t is in **transjunction** in a case c if and only if some input condition and some output condition of t are holding in the case c.

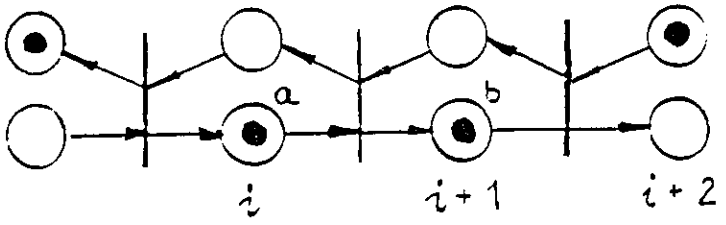
Formally:

$$\text{Transjunction } (t,c): \Longleftrightarrow {}^\bullet t \cap c \neq \emptyset \text{ and } t^\bullet \cap c \neq \emptyset$$

We have ruled out, in the above process of developing a secure net model for railway systems, three possible transjunction situations besides contact (forward as well as backward contact). Let us ask now why each of the three possibilities for transjunction has to be avoided:

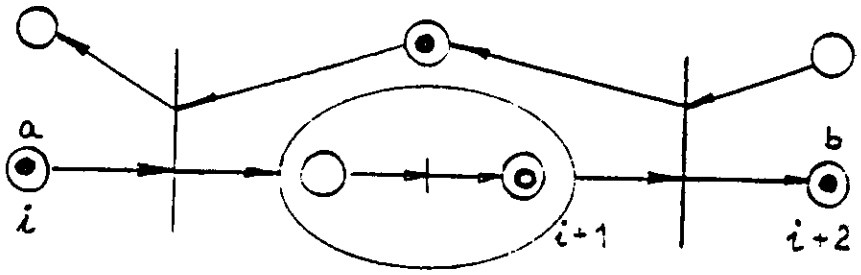
2.1 Train to Train Transjunction

The picture below illustrates such transjunction:



where in both segment i and segment $i+1$ there is a train (token): train a on segment i is not permitted to move now but train b on segment $i+1$ is permitted. If the body or tail

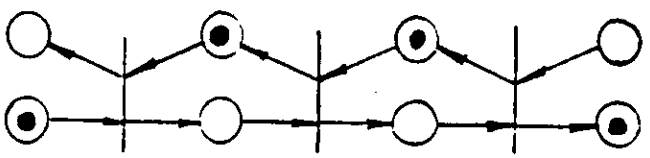
of train b is modelled explicitly, the situation should look like, after train b enters segment $i+2$, as below:



where, as shown, place $i+1$ has been decomposed and in one of the detailed inner places appears the train body (" "). Now, train a gets the permit signal to enter segment $i+1$. Within segment $i+1$, train a can move freely, or as fast as it likes to. Thus comes the danger: it may crash into the body of train b in case the latter is not moving quickly enough. So this type of transjunction reflects *insufficient separation* (of trains).

2.2 Signal to Signal Transjunction

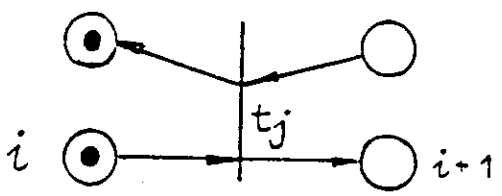
Pictorially, such a transjunction can be illustrated as below:



The danger is that when two permit signals "crash" into each other (as explained for train to train transjunction), chaos may occur, since they may merge and their identification may be impossible. Again, this type of transjunction reflects *insufficient separation* (of signals).

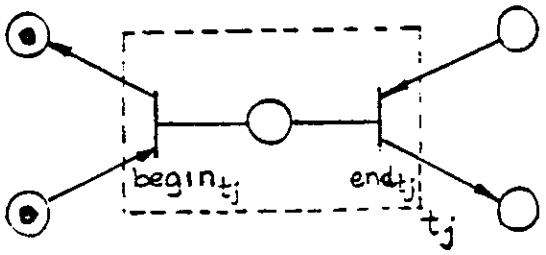
2.3 Train to Signal Transjunction

The following is a train to signal transjunction:

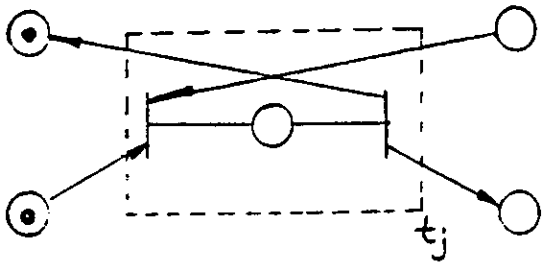


and it is not immediately clear why this possibility must be excluded, too. Transition t_j

in the picture may be decomposed (or implemented) as below:



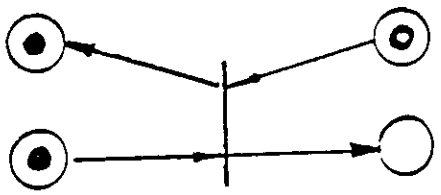
where a contact situation is detected around transition $begin_{t_j}$. Thus, a transjunction situation allows unsafe implementation. A proper implementation could be:



This decomposition of t_j can generate new output tokens only after having absorbed tokens from all inputs; therefore, if the net containing t_j is safe, it remains safe after this proper decomposition. Proper decomposition *preserves* safety (and security).

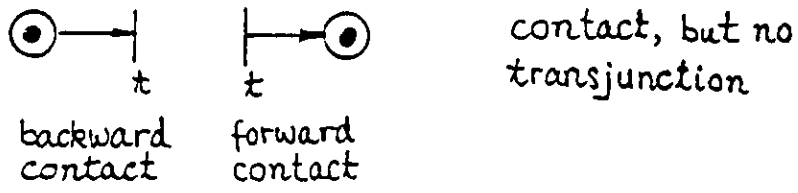
However, the possibility of unsafe decomposition is *not* the reason why we have to exclude train_to_signal transjunction! Unsafe decomposition may make even a secure net unsafe; it must be forbidden anyway.

Consider again train_to_signal transjunction t_j:



A signal token is shown in s; since even the shortest signal have some length, it is possible that part of the signal is still in s'. Therefore, the "door" t_j must open for that part of the signal, and thus it is open for the train. We have to keep in mind that transitions are like *signal* doors, open to all or to none. Therefore again, this type of transjunction reflects *insufficient separation*: between non_corresponding trains and permit signals. To sum up: all three transjunction situations should be avoided in a secure system, and therefore it is now apparent why security has been defined as no contact and no transjunction.

Note that "no transjunction" does not imply "no contact" since it may appear that ${}^\bullet t = \emptyset$ or $t^\bullet = \emptyset$ as illustrated by the pictures below:



As we all know, it is always possible, by S_complementation, to remove "contact" from a system, but it doesn't help in case of transjunction. The removal of transjunction requires to go into more system detail.

We have mentioned *backward contact* several times so far. We have the following two reasons to exclude (not only forward contact, but also) backward contact:

- If ${}^\bullet t \neq \emptyset \neq t^\bullet$ for some $t \in T$, then backward contact around t implies transjunction.
- Trains may move backwards.

So far we have talked about railway systems all the time. The difference between a train and a signal, as far as our model is concerned, lies in the mass property: it is nonzero for trains, but may be zero for permit signals. We have seen in section 1.3 that the properties and possible differences of trains and signals show up only in the *separation numbers*, n for trains and n' for permit signals. Both n and n' are ≥ 2 by definition. To achieve this separation, we need

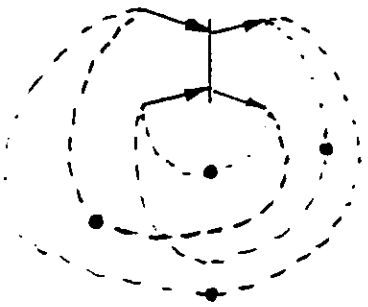
1. n backward lines
2. n' forward lines
3. the structural property of security

Otherwise, trains and permit signals can and must be treated alike.

Therefore, our net model $\Sigma_{2,2}$ describes also a *secure transmission line for messages* when all signals (now "message signals" and "permit signals" which may also be used for messages) are shorter than one segment: $n = n' = 2$. Such a transmission line has to be secure instead of merely safe since signals have nonzero length. Here is the security criterion:

- For marked directed graph (digraph for short):

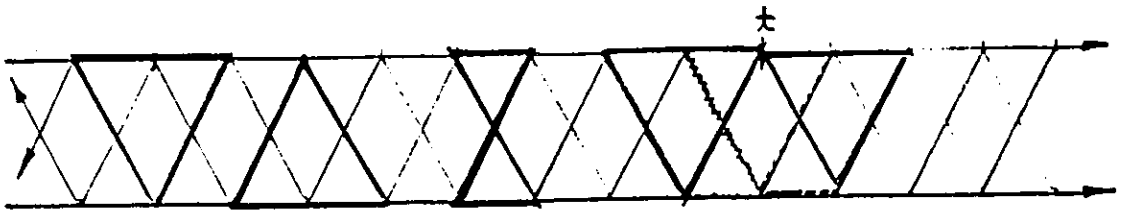
every *pair* of successive arcs must lie on a *basic circuit*:



A *basic circuit* is a (uniformly) directed circuit (mesh in net terminology) with exactly one edge marked.

- For (elementary) nets: Every transition with m inputs (preconditions) and n outputs (postconditions) must lie on $m \times n$ basic *domains* covering ${}^\bullet t^\bullet = {}^\bullet t \cup t^\bullet$, where ***domain*** is to be defined. Informally, a domain is a set of S_elements with a constant number of tokens; a basic domain is a domain where that constant is 1.

The following picture shows the three types of basic circuits for the simplest secure transmission line $\Sigma_{2,2}$, and the six basic circuits on which transition t lies:



3 Cycloids

Cycloids may serve as the basis for general $n \times k$ secure transmission lines. See next issue for an introductory discussion.