Title: Weaving Petri Nets
Author: Igor Benko
Date: January 26, 1996
Note #: IB-96-01

1 Motivation

Petri net based formalisms are commonly used for describing concurrent systems. For an introduction to Petri nets see, for example, [Peterson, 1981]. Various versions of Petri nets have been used for describing asynchronous systems; see [Dill et al., 1989] for an example of an application. Overview articles [Davis and Nowick, 1995] and [Brzozowski et al., 1995] give comprehensive lists of asynchronous design methodologies and describe a number methodologies based on Petri nets.

A useful technique for specifying complex behaviors, especially those that involve lots of concurrency, is employing a weave of snippets. Such an approach to the specification is often used in the domain of trace structures; see [Ebergen, 1992] for examples. Each snippet describes a small fraction of a behavior. The result of weaving the snippets is one snippet that corresponds to the conjunction of behaviors expressed by the individual snippets that were "woven".

When using Petri nets to describe snippets, we need the weave operator defined on Petri nets. In this report we define a Petri-net weave. Furthermore, we establish a formal relation between Petri nets and trace structures. Finally, we give a proof that, assuming the interleaving semantics for Petri net execution, the Petri-net weave is equivalent to the trace-structure weave.

We remark that Petri nets covered in this report allow for so-called open Petri nets, that is, Petri nets where a transition does not need to be connected to an input place or to an output place.

The Petri-net weave was, to the best of our knowledge, first used informally by Charles Molnar. Unfortunately, we were not able to trace an original reference.

2 Weaving

Figure 1 illustrates weaving. Snippet 1 describes the behavior where actions a and c alternate; snippet 2 describes the behavior where actions b and c alternate. The weaving of the two snippets results in snippet 3. Snippet 3 expresses the conjunction of the behaviors represented by snippets

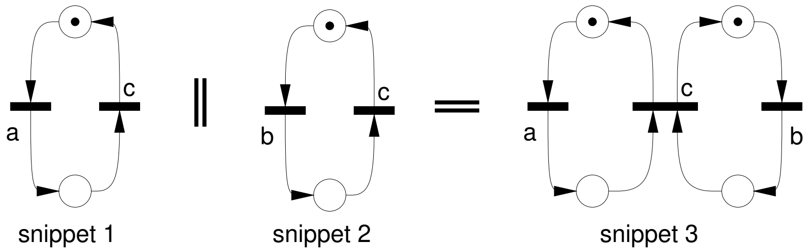

Figure 1: Snippet weave

1 and 2. Observe that in snippet 3 both actions a and b alternate with action c. Action c, however, cannot take place until allowed by behaviors of both snippet 1 and snippet 2. Sometimes we say that behaviors expressed in snippets 1 and 2 synchronize on action c.

3 Excerpts from bag theory

When dealing with Petri nets formally, we are going to use some basic bag theory. This section provides the background needed.

With $\mathbf{Bag}[\mathcal{D}]$ we denote a set of all bags over domain $\mathcal{D}$. A bag, like a set, is a collection of elements over some domain. However, unlike a set, bags allow *multiple* occurrence of elements. In set theory, an element is either a member of a set or not a member of a set. In bag theory, an element may be in a bag zero times (not in a bag), or one time, two times, or any specified number of times.

Here is an example of a bag over domain $\{a, b, c\}$:

$$B_0 = \{a, a, b, b\}$$

The basic notion in the bag theory is the number of occurrences of an element in a bag. For an element x and a bag B, we denote the number of occurrences of x in B with $\#(x, B)$, pronounced "the number of x in B".

An element x is a *member* of a bag B if the number of occurrences of x in B is larger than zero:

$$x \in B \equiv \#(x, B) > 0.$$

With $|B|$ we denote the cardinality of bag B:

$$|B| = \sum_{x \in \mathcal{D}} \#(x, B)$$

The table below summarizes the basic operations and relations on bags.

$$
\begin{array}{lrcl}
\textit{Bag inclusion:} & A \subseteq B & \equiv & \#(x,A) \leq \#(x,B) \\
\textit{Bag equality:} & A = B & \equiv & \#(x,A) = \#(x,B) \\
\textit{Bag union:} & \#(x, A \cup B) & = & \max(\,\#(x,A), \#(x,B)\,) \\
\textit{Bag intersection:} & \#(x, A \cap B) & = & \min(\,\#(x,A), \#(x,B)\,) \\
\textit{Bag sum:} & \#(x, A + B) & = & \#(x,A) + \#(x,B) \\
\textit{Bag difference:} & \#(x, A - B) & = & \#(x,A) - \#(x, A \cap B)
\end{array}
$$

4 Trace structures

Definition 4.1 (Trace structure) *A trace structure E is a pair $\langle \boldsymbol{a}.E, \boldsymbol{t}.E \rangle$, where $\boldsymbol{t}.E \subseteq \boldsymbol{a}.E^*$. $\boldsymbol{a}.E$ is the alphabet of the trace structure E, and $\boldsymbol{t}.E$ is the trace set of the trace structure E.*

The alphabet of the trace structure can be interpreted as the list of actions that can be taken by a device described by the trace structure.

Notice that in the definition of a trace structure above we did not make a distinction between inputs and outputs. Introducing such an extension is straightforward and is not relevant to the material of this report.

Before we list basic operations on trace structures, we recall a definition of the projection of a trace on a set of symbols.

Definition 4.2 (Projection)
Let Σ be a set of symbols, and let $t \in \Sigma^*$, $b \in \Sigma$, and $A \subseteq \Sigma^*$. The projection of trace t on alphabet A, denoted by $t \downarrow A$, is defined as follows:

$$
\begin{array}{lcll}
\varepsilon \downarrow A & = & \varepsilon & \\
(tb) \downarrow A & = & (t \downarrow A)b & \textit{if } b \in A \\
(tb) \downarrow A & = & t \downarrow A & \textit{if } b \notin A
\end{array}
$$

Intuitively, the projection $t \downarrow A$ deletes from t all symbols which are not in A. The projection extends naturally on sets of traces.

Table 1 summarizes the basic operations on trace structures; E and F are trace structures and A is a set of symbols.

Trace structure E, such that $E = \mathbf{pref}\, E$, is *prefix closed.*

5 Petri nets

We define a Petri net as a pair consisting of a Petri-net graph and a marking of that graph. The Petri-net graph defines the topology of a Petri net. The marking tells how tokens are distributed among the places of the Petri-net graph.

We begin our formalization of Petri nets by defining a Petri-net graph.

concatenation:	$E\,;F$	$=$	$\langle\, \boldsymbol{a}.E \cup \boldsymbol{a}.F,$	$(\boldsymbol{t}.E)(\boldsymbol{t}.F)\,\rangle$
union:	$E \mid F$	$=$	$\langle\, \boldsymbol{a}.R \cup \boldsymbol{a}.F,$	$\boldsymbol{t}.E \cup \boldsymbol{t}.F\,\rangle$
repetition:	$*[E]$	$=$	$\langle\, \boldsymbol{a}.E,$	$(\boldsymbol{t}.E)^*\,\rangle$
prefix closure:	$\mathbf{pref}\, E$	$=$	$\langle\, \boldsymbol{a}.E,$	$\{\, t \mid \exists s : ts \in \boldsymbol{t}.E\,\}\,\rangle$
projection:	$E \downarrow A$	$=$	$\langle\, \boldsymbol{a}.E \cap A,$	$\boldsymbol{t}.E \downarrow A\,\rangle$
weaving:	$E \parallel F$	$=$	$\langle \boldsymbol{a}.E \cup \boldsymbol{a}.F,$	$\{\, t \in (\boldsymbol{a}.E \cup \boldsymbol{a}.F)^* \mid t{\downarrow}\boldsymbol{a}.E \in \boldsymbol{t}.E \;\wedge\; t{\downarrow}\boldsymbol{a}.F \in \boldsymbol{t}.F\,\}\rangle$

Table 1: Operations on trace structures

Definition 5.1 (Petri-net graph)
A Petri net graph P is a triple $\langle \boldsymbol{a}.P, \mathbf{place}.P, \mathbf{trans}.P\rangle$ where:

- $\boldsymbol{a}.P$*, the alphabet of P, is a set of symbols.*
- $\mathbf{place}.P$ *is the set of places.*
- $\mathbf{trans}.P \subseteq \mathbf{Bag}[\mathbf{place}.P] \times \mathbf{Bag}[\mathbf{place}.P] \times (\boldsymbol{a}.P \cup \{\varepsilon\})$ *is the set of transitions.*

A transition itself is a triple consisting of a bag of input places, a bag of output places, and a transition label. Thus, the topology of a Petri-net graph is defined by means of transitions.

Definition 5.2 (Input places, output places, transition label)
Let $t = \langle \mathbf{ip}.t, \mathbf{op}.t, \mathbf{label}.t\rangle$ be a transition in Petri net graph P. $\mathbf{ip}.t$ is a bag of input places, $\mathbf{op}.t$ is a bag of output places, and $\mathbf{label}.t$ is the label of the transition.

$\mathbf{ip}.t$ and $\mathbf{op}.t$ are bags, because there may be more than one arc leading from a place to a transition or from a transition to a place. Notice that a bag may be empty, thus a transition does not have to be connected to an input place or to an output place.

In order to execute a Petri net, we place tokens in the places of a Petri-net graph. Each place can hold zero, one, two, or any specified number of tokens. A Petri net graph with tokens assigned to places is called a Petri net. In the following we use $\mathbb{N}$ to denote the set of natural numbers.

Definition 5.3 (Petri net)
A Petri net N is a pair $\langle \boldsymbol{g}.N, \boldsymbol{m}.N\rangle$ where $\boldsymbol{g}.N$ is a Petri-net graph, and $\boldsymbol{m}.N : \mathbf{place}.P \to \mathbb{N}$ is a marking of the Petri net graph $\boldsymbol{g}.N$.

Operators **a**, **place**, **trans** extend naturally to a Petri net.

In the context of specifying behavioral constraints by means of Petri nets, a Petri net is often called a snippet.

Definition 5.4 (Enabled transition)
Let $N = \langle P, Q\rangle$ be a Petri net. We say that transition t is enabled *by marking Q if for each incoming arc there is a distinct token in the place from which the arc originates:*

$$\mathbf{enabled}.(t, Q) \equiv (\forall p : p \in \mathbf{ip}.t : Q.p \geq \#(p, \mathbf{ip}.t)\,)$$

In the definition above we use $Q.p$ to denote the application of function Q to the argument p. Notice that a transition with no input places is always enabled.

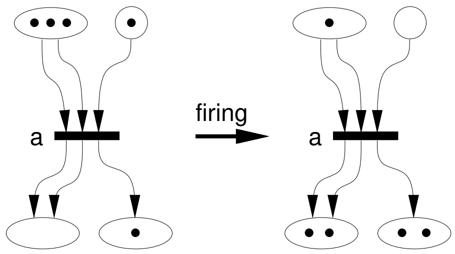

Figure 2: Firing a transition

An enabled transition may *fire*. Firing of a transition removes from each transition's input place one token for each arc that originates in that place and ends in the transition, and deposits in each transition's output place one token for each arc originating from the transition and ending in that place. Figure 2 shows an effect of firing a transition.

5.1 Interleaving semantics for Petri nets

We assume an interleaving semantics for Petri net execution, meaning that we allow only one enabled transition to fire at a time. We say that a Petri net fires with symbol a if a transition labeled with a fires. A firing can be seen as a transformation on the marking of the Petri net.

Definition 5.5 (Firing a symbol)
Let Q and Q' be markings of a Petri net graph P, $b \in \boldsymbol{a}.P \cup \{\varepsilon\}$. Firing of symbol b is a relation on markings of a Petri net graph defined as follows:

$$\begin{array}{lcl} Q \xrightarrow{\varepsilon}_P Q & \equiv & \text{true} \\ Q \xrightarrow{b}_P Q' & \equiv & \exists t \; : \; t \in \textbf{trans}.P \; : \; \Big(\textbf{enabled}.(t, Q) \;\wedge\; (\textbf{label}.t = b) \;\wedge \\ & & \qquad (\forall p \; : \; p \in \textbf{place}.P \; : \; Q'.p = Q.p - \#(p, \textbf{ip}.t) + \#(p, \textbf{op}.t)\,)\Big) \end{array}$$

Next we extend the firing relation on strings of symbols.

Definition 5.6 (Firing a string)
Let Q and Q' be markings of a Petri net graph P, $b \in \boldsymbol{a}.P \cup \{\varepsilon\}$, and s a possibly empty string on $\boldsymbol{a}.P$. Firing of string is a relation on markings of a Petri net graph, defined as follows:

$$\begin{array}{lcl} Q \xrightarrow{\varepsilon}{}^*_P Q & \equiv & \text{true} \\ Q \xrightarrow{sb}{}^*_P Q' & \equiv & \Big(\exists Q'' \; : \; Q'' \; \textit{a marking of} \; P \; : \; Q \xrightarrow{s}{}^*_P Q'' \;\wedge\; Q'' \xrightarrow{b}_P Q'\Big) \end{array}$$

Definition 5.7 (Trace set of a Petri net)
The trace set of Petri net N denoted by $\boldsymbol{t}.N$ is the set of all strings that N can fire from its initial marking:

$$\boldsymbol{t}.N = \left\{ s \mid s \in (\boldsymbol{a}.N)^* \ \wedge \ \exists Q \ : \ Q \text{ a marking of } P \,;\ \boldsymbol{m}.N \xrightarrow{s}{}^{*}_{P} Q \right\}$$

By definition, the trace set of a Petri net contains sequences of labels that correspond to all possible sequences of firings of the Petri net. Notice that the trace set of a Petri net is prefix closed.

Having a trace set of a Petri net we can establish a relation between Petri nets and trace structures.

Definition 5.8 (Trace structure of a Petri net)
The trace structure of a Petri net N, denoted by **trace**$.N$*, is the pair* $\langle \boldsymbol{a}.N, \boldsymbol{t}.N \rangle$.

6 Petri-net weave

Weave is an operation that is usually defined on trace structures. In this section we address the following problem:

> Given Petri nets M and N, construct Petri net R such that the trace structure of R is equal to the weave of the trace structures of M and N.

6.1 Weave intuitively

Intuitively, weaving is an operation used to construct a behavior that is a conjunction of behaviors described by snippets. Sometimes weaving is also called parallel "composition" with synchronization on common symbols. This characterization of the weaving is intended to emphasize the fact that in the composite behavior the symbols that are common to two snippets can be "fired" only if allowed by both snippets. If a symbol appears in the alphabet of one snippet only, the symbol may be fired when that snippet allows the firing.

When we construct the weave of Petri nets M and N we follow our intuitive interpretation of the weaving. Our intention is to let M and N run concurrently, and synchronize them on the common symbols. This means that any transition in M that is labeled with symbol b from $\boldsymbol{a}.M \cap \boldsymbol{a}.N$ may eventually synchronize with any of the transitions in N that are labeled with the same symbol b. Thus, each transition in $M \parallel N$ that is labeled with b must reflect the coupling of the operations of two Petri nets. In order to achieve this coupling, each transition t in $M \parallel N$ that is labeled with b is constructed from a pair of transitions t' from M and t'' from N. The intention is that t may fire exactly when both t' and t'' are enabled. Furthermore, after firing t, $M \parallel N$ must continue behaving like M and N running concurrently. Thus, input places of t are the union of input places of t' and t'', so t is enabled exactly when both t' and t'' are enabled. Output places of t are also the union of output places of t' and t'', so that after firing t the weave behaves as M and N running concurrently (with synchronization on common symbols).

The transitions in M and N that are labeled with symbols that are not common to M and N are transferred to $M \parallel N$ without a change, because these transitions are not a subject to synchronizations.

In order to preserve the information about enabled transitions, places $M \parallel N$ are exactly the union of places from M and N. The initial marking of places of $M \parallel N$ is the union of markings of M and N.

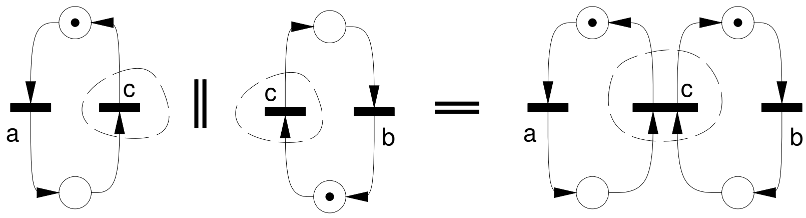

Figure 3: Weave example

Figure 3 repeats the scenario from Figure 1. Observe how encircled transitions on the left hand side are combined into the encircled transition on the right hand side.

Another example of a weave involving an open Petri net is shown in Figure 4.

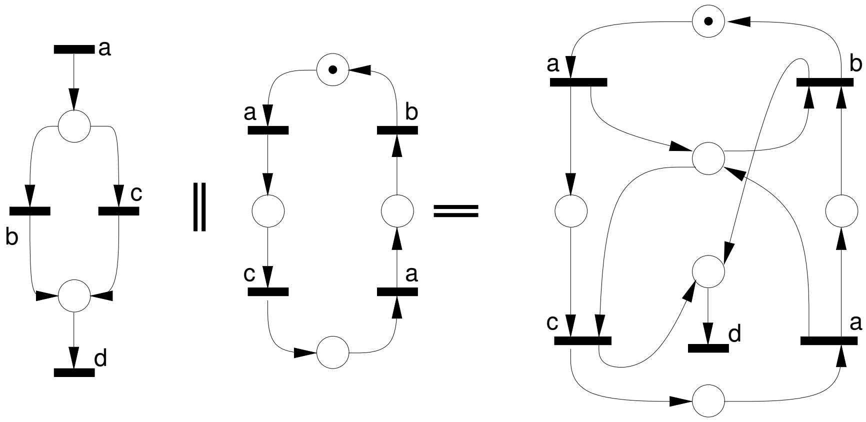

Figure 4: Another weave example

6.2 Weave formally

Now we define formally the weave of Petri nets. In the definition below $\div$ is used to denote the symmetric difference of sets. The symmetric difference of sets A and B is defined as

$$A \div B = (A \cup B) - (A \cap B)$$

Definition 6.1 (Petri net weave)
Let M and N be Petri nets such that $\mathbf{place}.M \cap \mathbf{place}.N = \emptyset$. *The weave of M and N, denoted by $M \parallel N$ is a Petri net and is defined as follows:*

- $\boldsymbol{a}.(M \parallel N) = \boldsymbol{a}.M \cup \boldsymbol{a}.N$
- $\mathbf{place}.(M \parallel N) = \mathbf{place}.M \cup \mathbf{place}.N$
- $$\begin{array}{rcl} \mathbf{trans}.(M \parallel N) & = & \{\, t \mid t \in (\mathbf{trans}.M \cup \mathbf{trans}.N) \ \wedge \ \mathbf{label}.t \in (\boldsymbol{a}.M \div \boldsymbol{a}.N) \cup \{\varepsilon\}\,\} \\ & \cup & \{\, \langle\, \mathbf{ip}.u \cup \mathbf{ip}.v,\ \mathbf{op}.u \cup \mathbf{op}.v,\ \mathbf{label}.u \,\rangle \mid u \in \mathbf{trans}.M \ \wedge \ v \in \mathbf{trans}.N \\ & & \quad \wedge\ \mathbf{label}.u = \mathbf{label}.v \,\} \end{array}$$
- $\boldsymbol{m}.(M \parallel N) = \boldsymbol{m}.M \cup \boldsymbol{m}.N$

Notice that the marking of a weave is defined as a union of two functions with *disjoint* domains. This observation is used in the proof we give for the theorem below.

The theorem below states that the Petri-net weave is equivalent to the weave of trace structures.

Theorem 6.2 (Weave equivalence)
Let M and N be Petri nets. Then

$$\mathbf{trace}.(M \parallel N) = \mathbf{trace}.M \parallel \mathbf{trace}.N$$

First we give a rather informal argument why the theorem above is correct, and then we prove the theorem formally.

6.2.1 An informal argument

By definition the alphabet of the Petri-net weave is equal to the union of the alphabets of the weavands. Thus, all we have to show is that the trace sets match.

In order to see that the two trace sets are equal we observe firing of transition a t of $M \parallel N$. Transition t fires if it is enabled, that is, if every input place p contains at least as many tokens as there are arcs from that place to transition t. The initial marking of $M \parallel N$ is a union of markings of M and N. First we determine which transitions are enabled initially. Let symbol b be a label of transition t. There are three cases we have to consider:

- $b \in \boldsymbol{a}.M - \boldsymbol{a}.N$. In this case all transitions that are labeled with b in $M \parallel N$ are also transitions in M. Because initially the marking of $M \parallel N$ is the union of markings of M and N, we conclude that t, such that $\textbf{label}.t = b$ is enabled in $M \parallel N$ if and only if it is enabled in M.

  Because input and output places of t in $M \parallel N$ match input and output places of t in M, we can also conclude that after firing transition t the marking of $M \parallel N$ is still the union of the marking of N and the marking of M after firing transition t.

- $b \in \boldsymbol{a}.N - \boldsymbol{a}.M$. This case is symmetrical to the case above.

- $b \in \boldsymbol{a}.N \cap \boldsymbol{a}.M$. Both input and output places of t can be partitioned in the places from $\textbf{place}.M$ and the places from $\textbf{place}.N$. Input places of t that are coming from M are also all input places of a transition t' in M such that $\textbf{label}.t' = b$. All output places of t' are also output places of t that came from M. Similar, input places of t that are coming from N are also all input places of a transition t'' in N such that $\textbf{label}.t'' = b$, and all output places of t'' are also output places of t that came from N. This means that initially transition t is enabled in $M \parallel N$ if and only if transitions t' and t'' are enabled.

  Because of the partitioning of the input and output places of t into the places coming from M and the places coming from N we conclude that the marking of $M \parallel N$ after firing transition t is the union of the marking of M after firing transition t' and the marking of N after firing transition t''.

From the reasoning above we conclude that string w of length 1 is in $\boldsymbol{t}.(M \parallel N)$ if and only if w is in $\boldsymbol{t}.M \parallel \boldsymbol{t}.N$.

Because trace sets corresponding to Petri nets are prefix-closed, we know that $\varepsilon \in \boldsymbol{t}.(M \parallel N)$ and $\varepsilon \in \boldsymbol{t}.M \parallel \boldsymbol{t}.N$.

We also conclude that after Petri net $M \parallel N$ fires transition t that is labeled with a symbol that is enabled by both M and N, the marking of $M \parallel N$ is the union of the markings of M and N after firing the transitions from which t was "composed".

Because after firing one symbol, the marking of $M \parallel N$ is again the union of the markings of M and N, we can apply the same argument to firing the next symbol. Applying the same argument to each subsequent firing leads to the conclusion that $\boldsymbol{t}.(M \parallel N) = \boldsymbol{t}.M \parallel \boldsymbol{t}.N$.

6.2.2 A formal argument

In the proof of the Theorem 6.2 we are going to employ induction on the length of the string. We are going to show that if and only if the weave of two Petri nets can fire a string, that is, the weave can do a sequence of firings, then the same sequence of firings can be performed by both weavands; firings of symbols that are unique to one weavand directly correspond to firings in that weavand, and firings of symbols that are common to both weavands correspond to firing of one transition in each weavand.

In order to make the proof of Theorem 6.2 more tractable, we introduce the following lemma:

Lemma 6.3
Let M and N be Petri nets, Q_1 and Q_1' markings of M, Q_2 and Q_2' markings of N, and $b \in \boldsymbol{a}.M \cup \boldsymbol{a}.N \cup \{\varepsilon\}$. Then the following equivalence holds

$$Q_1 \cup Q_2 \xrightarrow{b}_{\boldsymbol{g}.(M\|N)} Q_1' \cup Q_2' \quad \Longleftrightarrow \quad Q_1 \xrightarrow{b\downarrow \boldsymbol{a}.M}_{\boldsymbol{g}.M} Q_1' \ \wedge \ Q_2 \xrightarrow{b\downarrow \boldsymbol{a}.N}_{\boldsymbol{g}.N} Q_2'$$

The lemma states that a weave can fire a symbol if and only if both weavands can fire that symbol. Furthermore, the marking of the weave after firing a symbol is the union of the markings of the weavands after they have fired a symbol. The lemma is the cornerstone of our proof of Theorem 6.2, because it tells us that after firing of one symbol the tokens in the weave of two Petri nets are distributed in exactly the same way as the tokens in both weavands after they have fired the same symbol.

Proof: We have four cases to consider: $b \in \boldsymbol{a}.M - \boldsymbol{a}.N$, $b \in \boldsymbol{a}.N - \boldsymbol{a}.M$, $b \in \boldsymbol{a}.M \cap \boldsymbol{a}.N$, and $b = \varepsilon$.

Case 1: $b \in \boldsymbol{a}.M - \boldsymbol{a}.N$; or, $b = \varepsilon$, $Q_1 \neq Q_1'$ and $Q_2 = Q_2'$.

$$Q_1 \cup Q_2 \xrightarrow{b}_{\boldsymbol{g}.(M\|N)} Q_1' \cup Q_2'$$

$\Longleftrightarrow$ { Definition of $\longrightarrow$ }

$$\left(\exists t \ : \ t \in \mathbf{trans}.(M \parallel N) \ : \ \left(\mathbf{enabled}.(t, Q_1 \cup Q_2) \ \wedge \ \mathbf{label}.t = b \ \wedge \right.\right.$$
$$\left.\left. \left(\forall p \ : \ p \in \mathbf{place}.(M \parallel N) \ : \ (Q_1' \cup Q_2').p = (Q_1 \cup Q_2).p - \#(p, \mathbf{ip}.t) + \#(p, \mathbf{op}.t)\right)\right)\right)$$

$\Longleftrightarrow$ { Definition of weave; $b \notin \boldsymbol{a}.N \ \vee \ (b = \varepsilon \wedge Q_2 = Q_2') \implies t \in \mathbf{trans}.M$; $\mathbf{place}.M \cap \mathbf{place}.N = \emptyset$ } *

$$\left(\exists t \ : \ t \in \mathbf{trans}.M \ : \ \left(\mathbf{enabled}.(t, Q_1) \ \wedge \ \mathbf{label}.t = b \ \wedge \right.\right.$$
$$\left(\forall p \ : \ p \in \mathbf{place}.M \ : \ Q_1'.p = Q_1.p - \#(p, \mathbf{ip}.t) + \#(p, \mathbf{op}.t)\right) \ \wedge$$
$$\left.\left.\left(\forall p \ : \ p \in \mathbf{place}.N \ : \ Q_2'.p = Q_2.p\right)\right)\right)$$

$\Longleftrightarrow$ { Definition of $\longrightarrow$; $b = b \downarrow \boldsymbol{a}.M$, $\varepsilon = b \downarrow \boldsymbol{a}.N$, $Q_2 = Q_2'$ }

$$Q_1 \xrightarrow{b\downarrow \boldsymbol{a}.M}_{\boldsymbol{g}.M} Q_1' \ \wedge \ Q_2 \xrightarrow{b\downarrow \boldsymbol{a}.N}_{\boldsymbol{g}.N} Q_2'$$

Case 2: $b \in \boldsymbol{a}.N - \boldsymbol{a}.M$; or, $b = \varepsilon$, $Q_1 = Q_1'$ and $Q_2 \neq Q_2'$.
These two cases are symmetrical to Case 1.

Case 3: $b \in \boldsymbol{a}.N \cup \boldsymbol{a}.M$; or, $b = \varepsilon$, $Q_1 \neq Q_1'$ and $Q_2 \neq Q_2'$.

$$Q_1 \cup Q_2 \xrightarrow{b}_{\boldsymbol{g}.(M\|N)} Q_1' \cup Q_2'$$

$$\Longleftrightarrow \ \{ \text{ Definition of } \longrightarrow \ \}$$

$$\Big(\exists t \ : \ t \in \textbf{trans}.(M \parallel N) \ : \ \Big(\textbf{enabled}.(t, Q_1 \cup Q_2) \ \wedge \ \textbf{label}.t = b \ \wedge$$
$$\big(\forall p \ : \ p \in \textbf{place}.(M \parallel N) \ : \ (Q_1' \cup Q_2').p = (Q_1 \cup Q_2).p - \#(p, \textbf{ip}.t) + \#(p, \textbf{op}.t)\big)\Big)\Big)$$

$$\Longleftrightarrow \ \{ \text{ Definition of weave; } b \in \boldsymbol{a}.N \cup \boldsymbol{a}.M \text{ or } b = \varepsilon; \ \textbf{place}.M \cap \textbf{place}.N = \emptyset \ \} \qquad *$$

$$\Big(\exists u, v \ : \ u \in \textbf{trans}.M \ \wedge \ v \in \textbf{trans}.N \ :$$
$$\textbf{enabled}.(u, Q_1) \ \wedge \ \textbf{enabled}.(v, Q_2) \ \wedge \ \textbf{label}.u = \textbf{label}.v = b \ \wedge$$
$$\big(\forall p \ : \ p \in \textbf{place}.M \ : \ Q_1'.p = Q_1.p - \#(p, \textbf{ip}.t) + \#(p, \textbf{op}.t)\big) \ \wedge$$
$$\big(\forall p \ : \ p \in \textbf{place}.N \ : \ Q_2'.p = Q_2.p - \#(p, \textbf{ip}.t) + \#(p, \textbf{op}.t)\big)\Big)$$

$$\Longleftrightarrow \ \{ \text{ Definition of } \longrightarrow; \ b = b \downarrow \boldsymbol{a}.M = b \downarrow \boldsymbol{a}.N \ \}$$

$$Q_1 \xrightarrow{b \downarrow \boldsymbol{a}.M}_{\boldsymbol{g}.M} Q_1' \ \wedge \ Q_2 \xrightarrow{b \downarrow \boldsymbol{a}.N}_{\boldsymbol{g}.N} Q_2'$$

□

We remark that the steps marked with $*$ are the important steps where our definition of the Petri net weave is justified. Namely, when the weave fires a symbol that is unique to one weavand, the firing corresponds to firing of that weavand in isolation. When the weave fires a symbol that is common to both weavands, the firing corresponds to firing of one transition in each of the weavands.

Now we are ready to prove Theorem 6.2.

Proof: First we show the equality of alphabets. By definition,

$$\boldsymbol{a}.\textbf{trace}.(M \parallel N) = \boldsymbol{a}.(M \parallel N) = \boldsymbol{a}.M \cup \boldsymbol{a}.N = \boldsymbol{a}.\textbf{trace}.M \cup \boldsymbol{a}.\textbf{trace}.N = \boldsymbol{a}.(\textbf{trace}.M \parallel \textbf{trace}.N).$$

The next step is to prove

$$\boldsymbol{t}.(M \parallel N) \ = \ \boldsymbol{t}.N \parallel \boldsymbol{t}.N,$$

which is equivalent to

$$s \in \boldsymbol{t}.(M \parallel N) \quad \Longleftrightarrow \quad s \in (\boldsymbol{t}.M \parallel \boldsymbol{t}.N)$$

Instead of proving the equivalence above directly, we prove a slightly stronger statement, which implies the equivalence above. In the rest of the proof, Q_1 and Q'_1 are markings of Petri net M, and Q_2 and Q'_2 are markings of Petri net N. The statement we are going to prove is the following:

$$Q_1 \cup Q_2 \xrightarrow{s}{}^{*}_{\boldsymbol{g}.(M\|N)} Q'_1 \cup Q'_2 \quad \Longleftrightarrow \quad Q_1 \xrightarrow{s\downarrow \boldsymbol{a}.M}{}^{*}_{\boldsymbol{g}.M} Q'_1 \ \wedge \ Q_2 \xrightarrow{s\downarrow \boldsymbol{a}.N}{}^{*}_{\boldsymbol{g}.N} Q'_2$$

Observe that the left side of the equivalence above is, in isolation, equivalent to $s \in \boldsymbol{t}.M \parallel N$. The right side of the equivalence above is, in isolation, equivalent to $s \downarrow \boldsymbol{a}.M \in \boldsymbol{t}.M \ \wedge \ s \downarrow \boldsymbol{a}.N \in \boldsymbol{t}.N$.

We carry out the proof by induction on the length of the string. We use the following expression:

$$H(n) = \left(\forall s \ : \ |s| = n \ : Q_1 \cup Q_2 \xrightarrow{s}{}^{*}_{\boldsymbol{g}.(M\|N)} Q'_1 \cup Q'_2 \iff Q_1 \xrightarrow{s\downarrow \boldsymbol{a}.M}{}^{*}_{\boldsymbol{g}.M} Q'_1 \ \wedge \ Q_2 \xrightarrow{s\downarrow \boldsymbol{a}.N}{}^{*}_{\boldsymbol{g}.N} Q'_2\right)$$

Induction basis: First we assume $n = 0$. The only choice for s is then $s = \varepsilon$. By the definition of firing a string

$$Q_1 \cup Q_2 \xrightarrow{\varepsilon}{}^{*}_{\boldsymbol{g}.(M\|N)} Q_1 \cup Q_2$$

is always true, and similarly

$$Q_1 \xrightarrow{\varepsilon}{}^{*}_{\boldsymbol{g}.M} Q_1 \ \wedge \ Q_2 \xrightarrow{\varepsilon}{}^{*}_{\boldsymbol{g}.N} Q_2$$

is always true. Thus, H(0) holds.

Induction hypothesis: We take $n \geq 0$, and we assume that $H(k)$ holds for all k, such that $k \leq n$.

Induction step: Consider string sb, such that $|s| = n$ and $b \in \boldsymbol{a}.M \cup \boldsymbol{a}.N$.

$$Q_1 \cup Q_2 \xrightarrow{sb}{}^{*}_{\boldsymbol{g}.(M\|N)} Q'_1 \cup Q'_2$$

$\Longleftrightarrow$ { Definition of $\longrightarrow^{*}$ }

$$\begin{aligned} &\exists Q''_1, Q''_2 \ : \ (Q''_1 \text{ a place in } M) \ \wedge \ (Q''_2 \text{ a place in } N) \ : \\ &\qquad Q_1 \cup Q_2 \xrightarrow{s}{}^{*}_{\boldsymbol{g}.(M\|N)} Q''_1 \cup Q''_2 \ \wedge \ Q''_1 \cup Q''_2 \xrightarrow{b}_{\boldsymbol{g}.(M\|N)} Q'_1 \cup Q'_2 \end{aligned}$$

$\Longleftrightarrow$ { Induction hypothesis }

$$\begin{aligned} &\exists Q''_1, Q''_2 \ : \ (Q''_1 \text{ a place in } M) \ \wedge \ (Q''_2 \text{ a place in } N) \ : \\ &\quad Q_1 \xrightarrow{s\downarrow \boldsymbol{a}.M}{}^{*}_{\boldsymbol{g}.M} Q''_1 \ \wedge \ Q_2 \xrightarrow{s\downarrow \boldsymbol{a}.N}{}^{*}_{\boldsymbol{g}.N} Q''_2 \\ &\quad Q''_1 \cup Q''_2 \xrightarrow{b}_{\boldsymbol{g}.(M\|N)} Q'_1 \cup Q'_2 \end{aligned}$$

$\Longleftrightarrow$ { Lemma 6.3 }

$$\begin{aligned} &\exists Q''_1, Q''_2 \ : \ (Q''_1 \text{ a place in } M) \ \wedge \ (Q''_2 \text{ a place in } N) \ : \\ &\quad Q_1 \xrightarrow{s\downarrow \boldsymbol{a}.M}{}^{*}_{\boldsymbol{g}.M} Q''_1 \ \wedge \ Q_2 \xrightarrow{s\downarrow \boldsymbol{a}.N}{}^{*}_{\boldsymbol{g}.N} Q''_2 \\ &\quad Q''_1 \xrightarrow{b\downarrow \boldsymbol{a}.M}_{\boldsymbol{g}.M} Q'_1 \ \wedge \ Q''_2 \xrightarrow{b\downarrow \boldsymbol{a}.N}_{\boldsymbol{g}.N} Q'_2 \end{aligned}$$

$\Longleftrightarrow$ { Definition of $\longrightarrow^*$ }

$$Q_1 \xrightarrow{sb\downarrow a.M}{}^*_{g.M} Q_1' \;\wedge\; Q_2 \xrightarrow{sb\downarrow a.N}{}^*_{g.N} Q_2'$$

Thus, $H(n)$ holds for all n, such that $n \geq 0$. $\Box$

7 Conclusions

In the report we established a formal relation between Petri nets and trace structures and showed how weaving, a useful operation on trace structures, can be defined on Petri nets.

We gave an informal argument and a formal proof that the Petri-net weave is equivalent to the trace-structure weave. The formal proof provides an insight into the particulars of our definition of the Petri-net weave.

Finally, we remark that the formalization of Petri nets and the definition of the weave given in this report also apply to so-called open Petri nets.

References

[Brzozowski et al., 1995] Brzozowski, J. A., Hauck, S., and Seger, C.-J. H. (1995). Design of asynchronous circuits. In Brzozowski, J. A. and Seger, C.-J. H. *Asynchronous Circuits, Chapter 15.* Springer-Verlag.

[Davis and Nowick, 1995] Davis, A. and Nowick, S. M. (1995). Asynchronous circuit design: Motivation, background, and methods. In Birtwistle, G. and Davis, A., editors, *Asynchronous Digital Circuit Design*, Workshops in Computing, pages 1–49. Springer-Verlag.

[Dill et al., 1989] Dill, D. L., Nowick, S. M., and Sproull, R. F. (1989). Automatic verification of speed-independent circuits with Petri net specifications. In *Proc. International Conf. Computer Design (ICCD)*, pages 212–216. IEEE Computer Society Press.

[Ebergen, 1991] Ebergen, J. C. (1991). A Formal Approach to Designing Delay-Insensitive Circuits. *Distributed Computing*, 5(3):107–119.

[Ebergen, 1992] Ebergen, J. C. (1992). Arbiters: an exercise in specifying and decomposing asynchronously communicating components. *Science of Computer Programming*, 18(3):223–245.

[Molnar et al., 1985] Molnar, C. E., Fang, T.-P., and Rosenberger, F. U. (1985). Synthesis of delay-insensitive modules. In Fuchs, H., editor, *1985 Chapel Hill Conference on Very Large Scale Integration*, pages 67–86. Computer Science Press.

[Peterson, 1981] Peterson, J. L. (1981). *Petri Net Theory and the Modeling of Systems.* Prentice-Hall.