

Notes on Petri Nets and Attack Trees

Aubrey Bryant and Harley Eades III

July 5, 2018

Abstract

TODO

1 Petri Nets

Definition 1. *Petri net: a 5-tuple (P, T, F, V, m_0)*

P = a set of places; like states in automata; denoted by circles or vertices of one color

T = a set of transitions; denoted by rectangles or bars

F = the flow relation, a set of arcs; denoted by directional lines

V = multiplicity function mapping the weight of the arcs (this indicates how many tokens are needed to activate the arc)

M_0 = the initial marking that assigns a natural number to each place, roughly like resources that flow through the net

tokens: a discrete number of marks denoted as dots inside the places

these track the execution of the petri net - show what 'instruction' it is on, or what state it is in

Restrictions:

P and T are finite sets. $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.

F is a relation between P and T : $F \subseteq (P \times T) \cup (T \times P)$

$V: F \rightarrow \mathbb{N}^+$

Firing of a petri net: this captures the dynamic behavior of the petri net

Firing enables the transition from the initial marking to successor markings based on the firing rules

Pre-place: an input place; a place with an arc to a transition; denoted (\cdot, t) or $\cdot t$

If a pre-place has at least as many marks as the output arc's weight, then it is fulfilled and the transition the arc points to is activated, causing the transition to output to the post-place. However, the transition outputs the weight of its output arcs. Output \neq Input

Post-place: an output place; a place with an arc from a transition; denoted (t, \cdot) or $t \cdot$

A firing sequence from a place is a trace through a sequence of transitions

A reachable marking from one place to another means that there is a firing sequence from the first place to the last

The set $RN := RN(m_0)$ is the state space, the set of all reachable states in the petri net

This holds information about what events are possible and impossible in the petri net

Bounded: a petri net is bounded if its state space is finite

A k-safe petri net is one in which, in every marking reachable from the initial marking, there are at most k tokens.

In classic petri nets, aka 1-safe petri nets, places can have at most one mark

1.1 Pomset/Petri Net equivalence

Since pomsets do not have a way to account for tokens or weighted arcs, we need to focus on 1-safe petri nets ; Best and Wimmel (2000) have a method for converting k-safe petri nets to 1-safe petri nets.

The arcs of a pomset need to represent the direct predecessor relation only.

Best and Wimmel (2000) explain that a pomset is an abstraction from a process, which is based on an occurrence net:

A process is a tuple (B, E, F, r) where $r: (B \cup E) \rightarrow (S \cup T)$ and:

(B, E, F) is an occurrence net:

An occurrence net is a 1-safe petri net with some additional restrictions. (Hayman and Winskel 2008)

B is a set of conditions s.t. $(r(B) \subseteq S)$; these describe the state of a place

(To 'unfold' a net is to make each implicit option or path explicit; to draw each path as a separate thread. Thus, conditions describe the state of a place, or 'what's inside' the place).

E is a set of events s.t. $(r(E) \subseteq T)$

F is a function which can be viewed as a relation on $(B \times E) \cup (E \times B)$ - describes ordering of conditions and events

$\min O = b \in B | F(., b) = \emptyset$ and $\max O = b \in B | F(b, .) = \emptyset$

Conditions for an O-net:

1) $\forall x \in B \cup E$, the set $F(x)$ is a singleton. (this means there is only one path from a b to an e or e to b .)

2) The relation F is acyclic - the transitive closure of F is irreflexive

3) $\forall b \in B : |F(., b)| \leq 1$ and $|F(b, .)| \leq 1$. There is one or fewer paths into or out of a b .

Hayman and Winskel (2008) describe this as F being a "flow relation describing how places and transitions are connected" and F not being a multirelation.

(Best and Wimmel 2000)

Hickmott et al. (2007) describe the process of unfolding a Petri net into an occurrence net. (An occurrence is the same thing as a firing of the net.)

The occurrence net makes explicit all the possible runs of the PT net from the initial marking.

The particular occurrences of the places and transitions of a PT-net are represented as conditions and events in O-nets.

Unfolding makes the net simpler by making multiple instances of places as needed, hiding unreachable places, and eliminating cycles and backward conflicts. A backward conflict occurs when two transitions output to the same place, making it impossible to know which one actually fired. Eliminating backward conflict gives the action set post-uniqueness, meaning that we know the precise set of actions that brought about a certain marking (i.e. state of affairs). Forward conflict happens when two transitions are enabled but only one can fire (because two transitions branch off of one place), a situation that results in the same ambiguity observed with backward conflict.

(Unfolding of a Petri-net can often be infinite, but at some point the unfolding can be stopped without loss of information. This stopped unfolding is called a complete finite prefix.)

Configurations represent possible partial runs of the Petri net, and meet two conditions:

- 1) Causally closed: if any event is in the configuration, so are its ancestors: $\forall e \leq e, e \in C \implies e \in C$.
- 2) No forward conflict.

Additionally, configurations can be associated with a marking by identifying the marking that will result once that configuration is fired (starting from the initial marking).

Unfolding:

A homomorphism from the O-net to the Petri-net maps conditions B and events E onto places P and transitions T.

One place can be mapped into multiple conditions, to meet the conditions of configurations above and show different possible runs.

Transitions that are unreachable under any possible marking of the PT-net disappear in the O-net, bc they will never occur.

The O-net starts by mapping conditions with empty presets onto places initially marked in the PT-net.

Then, each possible firing is explored, and each possible run is added as a set of conditions and events, until cut-off events are reached in each thread. (Cut off events are those past which no new information is added, and thus represent the limit of the complete finite prefix).

The unfolding duplicates nodes as needed to guarantee post-uniqueness - meaning

each condition will have a unique event as a predecessor, no sharing of predecessors.

Best et al. (2007) explain that the pomset language of (N, λ) is "the set of λ -images of finite processes of N ." Where N is a net and λ is a transition labeling.

2 Notes on Attack Trees and Petri Nets

Definition 2. Graph G : a 4-tuple (T, V, src, tar) where

T is a set of edges;

V is a set of nodes;

src is $T \rightarrow V$ is the source of a given edge;

tar is $T \rightarrow V$ is the target of a given edge.

Definition 3. A graph morphism, $h : G \rightarrow G'$, between graphs is a pair (f, g) where:

$f: T \rightarrow T'$ is a function, and

$g: V \rightarrow V'$ is a function.

Definition 4. A finite multiset over a set S can be defined as a formal sum:

$$n_1 a_1, \oplus \dots \oplus, n_i a_i$$

where $n_1, \dots, n_i \in \mathbb{N}$, and $a_1, \dots, a_i \in S$

The formal sum obeys:

- na : a occurs n times in multiset,
- order doesn't matter,
- $na \oplus ma = (m+n)a$,
- $na \oplus 0 = na = 0 \oplus na$.

Definition 5. The free commutative monoid generated by the set A is:

$$A^\oplus = \{m \mid m \text{ is a finite multiset over } A\}.$$

Definition 6. A place-transition Petri Net is a graph, (T, S^\oplus) where T is called the set of transitions and S is called the set of places.

The elements of S^\oplus model the number of markings required for a transition to fire at a particular place.

Definition 7. A monoid homomorphism between two monoids $(M_1, \oplus_1, 0_1)$ and $(M_2, \oplus_2, 0_2)$ is a function $f : M_1 \rightarrow M_2$ such that:

- $f(a_1 \oplus_1 a_2) = f a_1 \oplus f a_2$
- $f 0_1 = 0_2$

Definition 8. A Petri Net morphism between two Petri nets (T_1, S_1^\oplus) and (T_2, S_2^\oplus) is a graph morphism (f, g) where $g : S_1^\oplus \rightarrow S_2^\oplus$ is a monoid homomorphism.

Thus, take as objects Petri nets (T, S^\oplus) and as morphisms Petri net morphisms. This defines a category \mathbf{Petri} .

Lemma 9. *Suppose we have two Petri nets, $(S_1, \oplus, 0)$ and $(S_2, \oplus, 0)$. The product of their free commutative monoids is isomorphic to the free commutative monoid of their disjoint union. In symbols,*

$$(S_1^\oplus \times S_2^\oplus) \cong (S_1 + S_2)^\oplus$$

To show this, we need to define two homomorphisms between them, which we will call F and G .

Proof. It is important to note that the \oplus operation is commutative. This commutativity allows the elements of the resulting free commutative monoid (FCM) to be rearranged, which will help us in our proof by allowing FCMs to be sorted in the following way:

Consider an arbitrary $s \in (S_1 + S_2)^\oplus$. Since the generator is the disjoint union of S_1 and S_2 , the elements of s are drawn from those two sets, but are not grouped according to their origin set. Using commutativity, we can move all $y_i \in S_1$ to the front, and all $z_j \in S_2$ to the end. s then has the form $(\oplus_i y_i) \oplus (\oplus_j z_j)$, where the elements are grouped according to their origin set. Any element of $(S_1 + S_2)^\oplus$ can be similarly sorted. For the purposes of this proof, assume that all such elements are so sorted.

Now, let us define the function $F: (S_1 + S_2)^\oplus \rightarrow (S_1^\oplus \times S_2^\oplus)$.

Let k be an arbitrary element of $(S_1 + S_2)^\oplus$. We know that $k \in (S_1 + S_2)^\oplus$ must have the form $(\oplus_i y_i) \oplus (\oplus_j z_j)$, where $y_i \in S_1$ and $z_i \in S_2$ as described above. Furthermore, since the $+$ operator is disjoint union, each element includes a marker indicating its origin set, which we can use to locate the boundary between elements of S_1 and elements of S_2 . Therefore we can define F as follows:

$$F((\oplus_i y_i \in S_1) \oplus (\oplus_j z_j \in S_2)) = ((\oplus_i y_i \in S_1), (\oplus_j z_j \in S_2))$$

$$F((S_1, \oplus) \oplus (S_2, \oplus)) = ((S_1, \oplus), (S_2, \oplus))$$

This allows an element of $(S_1 + S_2)^\oplus$ to be matched with an element of $(S_1^\oplus \times S_2^\oplus)$, since the generator of $(S_1 + S_2)^\oplus$ contains the generator for both S_1^\oplus and S_2^\oplus .

Another important feature of FCMs arises from lifting the coproducts of sets to FCMs. For the coproduct of sets, we know that there exist the following functions:

$$inj_1 : S_1 \rightarrow (S_1 + S_2)$$

$$inj_2 : S_2 \rightarrow (S_1 + S_2)$$

Likewise, for the coproduct of FCMs, there are the functions:

$$inj_1^\oplus : S_1^\oplus \rightarrow (S_1 + S_2)^\oplus : inj_1^\oplus(\oplus_i n_i x_i) = \oplus_i n_i (inj_1 x_i)$$

$$inj_2^\oplus : S_2^\oplus \rightarrow (S_1 + S_2)^\oplus : inj_2^\oplus(\oplus_i n_i x_i) = \oplus_i n_i (inj_2 x_i)$$

Using this property, let us now define $G : (S_1^\oplus \times S_2^\oplus) \rightarrow (S_1 + S_2)^\oplus$.

Let k be an arbitrary element of $(S_1^\oplus \times S_2^\oplus)$. We know k must have the form (M_1, M_2) where M_i is a multiset of the form $(n_1 z_1 \oplus \dots \oplus n_b z_b)$, and all $z \in S_i$ for $i \in \{1, 2\}$ (all n being natural number counters for the multiset).

Therefore we define the function G as follows:

$$G(M_1, M_2) = (in_{j_1}^\oplus M_1) \oplus (in_{j_2}^\oplus M_2)$$

By this transformation, we can match an element of $(S_1^\oplus \times S_2^\oplus)$ with an element of $(S_1 + S_2)^\oplus$, since by the nature of coproducts, any $s \in S_1^\oplus \in (S_1 + S_2)^\oplus$, and similarly any $s \in S_2^\oplus \in (S_1 + S_2)^\oplus$.

Now that we have defined a homomorphism in both directions, we must prove that $F;G = \text{Id}$ and $G;F = \text{Id}$. Take an arbitrary multiset M of the form $(n_1 z_1 \oplus \dots \oplus n_b z_b)$, with some elements drawn from a set S_a and some drawn from S_b (all n being natural number counters for the multiset). Sorting this multiset will yield the form $(\oplus_a n_a y_a) \oplus (\oplus_b n_b z_b)$, where $y_a \in S_a$ and $z_b \in S_b$. Putting this into the functions, we get:

$$F((\oplus_a n_a y_a) \oplus (\oplus_b n_b z_b)) = ((\oplus_a n_a y_a), (\oplus_b n_b z_b))$$

$$G((\oplus_a n_a y_a), (\oplus_b n_b z_b)) = ((\oplus_a n_a y_a) \oplus (\oplus_b n_b z_b))$$

Thus, $F;G = \text{Id}$.

Now let us check for identity in the opposite direction, proving $G;F = \text{Id}$.

Take arbitrary sets S_a and S_b . Generate the FCM of each and then their product: $(S_a^\oplus \times S_b^\oplus)$. The result will have the form $((\oplus_a n_a y_a), (\oplus_b n_b z_b))$, where $y_a \in S_a$ and $z_b \in S_b$. Putting this into the functions, we get:

$$G((\oplus_a n_a y_a), (\oplus_b n_b z_b)) = ((\oplus_a n_a y_a) \oplus (\oplus_b n_b z_b))$$

$$F((\oplus_a n_a y_a) \oplus (\oplus_b n_b z_b)) = ((\oplus_a n_a y_a), (\oplus_b n_b z_b))$$

Thus, $G;F = \text{Id}$. Since $(S_1^\oplus \times S_2^\oplus) \leftrightarrow (S_1 + S_2)^\oplus$, we can conclude that $(S_1^\oplus \times S_2^\oplus) \cong (S_1 + S_2)^\oplus$.

□

Definition 10. A chainable petri net is a Petri net $N = (\delta_0, \delta_1 : T \rightarrow S^\oplus)$ with two additional features: an element $i \in S^\oplus$ that points to the initial marking, and an element $f \in S^\oplus$ that is the endpoint of the final marking. $\text{tar}(i) = (s_1 \oplus s_2 \dots \oplus s_n)$, with elements not duplicated: $s_j \neq s_k$ when $j \neq k$. i points to the initial marking of the net, and so it gives the starting point for the net. Thus, $\text{src}(i) = \emptyset$.

The element f tracks the ending of the net, so $\text{src}(\{x \in \text{final}\}) = f$, and $\text{tar}(f) = \emptyset$. Morphisms for chainable nets must preserve both i and f . So, a morphism $\langle a, b \rangle : (N, i, f) \rightarrow (N', i', f')$ is an ordinary net morphism that preserves the markings - $b(i) = i'$ and $b(f) = f'$.

The chainable petri net, or CNET, is a petri net that tracks both its initial and its final places, to facilitate chaining the petri net together with other petri nets at either

end. This enables the use of operators such as SEQ, AND, & OR.

Suppose we have two CNETs, (N_1, i_1, f_1) and (N_2, i_2, f_2) .

$N_1 = (\delta_0, \delta_1 : T \rightarrow S)$ and $N_2 = (\delta'_0, \delta'_1 : T' \rightarrow S')$.

SEQ for CNETs is the composition operation:

$N_1 \text{ SEQ } N_2 = N_1 ; N_2 = N_3$, where (N_3, i_3, f_3) such that:

$N_1 ; N_2 = T_1 \cup T_2 \rightarrow S_1^\oplus \cup S_2^\oplus$;

$i_3 = i_1$;

$f_3 = f_2$;

$\text{tar}(f_1) = i_2$;

$\text{src}(i_2) = f_1$.

AND for CNETs is the categorical product(see pg. 114):

$N_1 \text{ AND } N_2 = N_1 \times N_2 = N_3$, where (N_3, i_3, f_3) such that:

$N_1 \times N_2 = T_1 \times T_2 \rightarrow S_1^\oplus \times S_2^\oplus$;

i_3 is a new place;

$\text{src}(i_3 \times \emptyset) = \emptyset$;

$\text{tar}(i_3 \times \emptyset) = \{i_1 \oplus i_2\}$;

f_3 is a new place;

$\text{src}(f_3 \times \emptyset) = \{f_1 \oplus f_2\}$;

$\text{tar}(f_3 \times \emptyset) = \emptyset$;

OR for CNETs is the coproduct(p.126):

$N_1 \text{ OR } N_2 = N_3$, where (N_3, i_3, f_3) such that:

$N_1 + N_2 = T_1 + T_2 \rightarrow (S_1 + S_2)^\oplus$;

i_3 is a new place;

$\text{src}(i_3) = \emptyset$;

$\text{tar}(i_3) = \{i_1 \oplus i_2\}$;

f_3 is a new place;

$\text{src}(f_3) = \{f_1\}$;

$\text{src}(f_3) = \{f_2\}$;

$\text{src}(f_3) = \{f_1 \oplus f_2\}$;

$\text{tar}(i_3) = \emptyset$

Example nets: $K =$

$S_K : \{a, b, c\}$

$T_K : \{t\}$

$F_K(a, t) = 2$

$F_K(b, t) = 1$

$F_K(t, c) = 2$

$F_K(\text{else}) = 0$

$S_K^\oplus : \{2a \oplus 1b \oplus 2c\}$

$t = \{2a \oplus 1b \rightarrow 2c\}$

$\delta_{0K}(t) = 2a \oplus 1b\}$

$\delta_{1K}(t) = 2c\}$

$M =$

$$\begin{aligned}
S_M &: \{d, e\} \\
T_M &: \{t'\} \\
F_M(d, t') &= 2 \\
F_M(t', e) &= 1 \\
F_M(\text{else}) &= 0 \\
S_M^\oplus &: \{2d \oplus 1e\} \\
t &= \{2d \rightarrow 1e\} \\
\delta_{0M}(t') &= 2d \\
\delta_{1M}(t') &= 1e \\
S_K^\oplus \times S_M^\oplus &= (2a \oplus 1b, 2d) \rightarrow (2c, 1e) \\
(S_K + S_M)^\oplus &= (\{1\} \times S_K) \cup (\{2\} \times S_M)^\oplus \\
((1, 2a), (1, 1b), (2, 2d))^\oplus &= \\
S_{KM}^\oplus &: \{(1, 2a), (1, 1b), (2, 2d)\} \\
T_{KM} &: \{t_1, t_2\} \\
t_1 &= ((1, 2a) \oplus (1, 1b)) \rightarrow (1, 2c) \\
t_2 &= (2, 2d) \rightarrow (2, 1e) \\
\delta_{0KM}(t_1) &= (1, 2a) \oplus (1, 1b) \\
\delta_{0KM}(t_2) &= (2, 2d) \\
\delta_{1KM}(t_1) &= (1, 2c) \\
\delta_{1KM}(t_2) &= (2, 1e) \\
(S_1^\oplus \times S_2^\oplus) \text{ and } (S_1 + S_2)^\oplus &\text{ are isomorphic.}
\end{aligned}$$

Suppose there are two Petri nets S_1 and S_2 , with sets of places M_1 and M_2 and sets of arcs T_1 and T_2 .

Let F be the homomorphism from $(S_1 + S_2)^\oplus$ to $(S_1^\oplus \times S_2^\oplus)$.

$$\begin{aligned}
F((S_1 + S_2)^\oplus) &= (S_1^\oplus + S_2^\oplus) \\
\text{where } S_1 + S_2 &= (\{1\} \times S_1) \cup (\{2\} \times S_2)
\end{aligned}$$

For F :

Calculating the set of places of the codomain:

$$\begin{aligned}
S_1 &= \bigcup_{y \in \text{domain}(x, y)} |x = 1) \\
S_2 &= \bigcup_{y \in \text{domain}(x, y)} |x = 2)
\end{aligned}$$

Note: I mean to separate out the two sets using the disjoint markers here.

The set of places of the codomain = $(S_1^\oplus \times S_2^\oplus)$

The arrows of the codomain maintain the structure defined in the domain, disregarding the origin markers $\{1\}$ and $\{2\}$ added in by taking the disjoint union.

Since the function preserves the structure between the objects, only changing the names of the objects, this is a homomorphism.

Now, let G be the homomorphism from $(S_1^\oplus \times S_2^\oplus)$ to $(S_1 + S_2)^\oplus$.

The objects or places of the codomain are the objects of the domain modified in the following way:

where objects in the domain have the form $(x, y), (\{1\} \times x) \cup (\{2\} \times y)$.

The arrows of the codomain maintain the structure of the domain, adding in origin markers $\{1\}$ and $\{2\}$ by position as described above.

Since G preserves the group's structure and only modifies objects' names, it is a homomorphism. Clearly, $G \circ F = F \circ G$, since F removes the origin markings $\{1\}$ and $\{2\}$ and G puts them back. Structure remains unchanged in either direction.

References

- [1]
- [2] Mario Sáenz Espinoza, Jose Goncalves, Paulo Leitao, Jose Luis Gonzalez Sanchez, and Alberto Herreros. Inverse kinematics of a 10 dof modular hyper-redundant robot resorting to exhaustive and error-optimization methods: A comparative study. In *Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012 Brazilian*, pages 125–130. IEEE, 2012.
- [3] Mario Sáenz Espinoza, Ana I Pereira, and José Gonçalves. Optimization methods for hyper-redundant robots’ inverse kinematics in biomedical applications. In *AIP Conference Proceedings*, volume 1479, page 818, 2012.
- [4] Ming Tham. Writing research theses or dissertations, May 2001. University of Newcastle Upon Tyne.