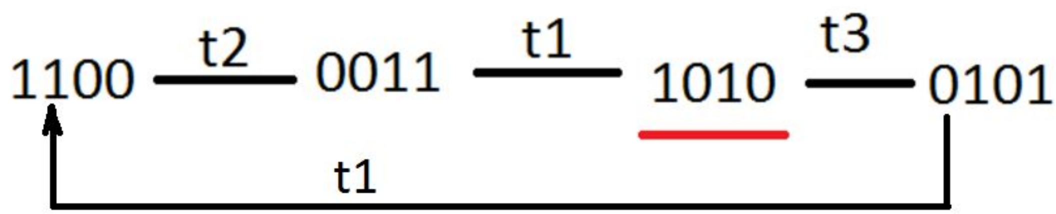
**Anton Yeshchenko**
**Kerwin Jorbina**

## Part B. Shortest firings

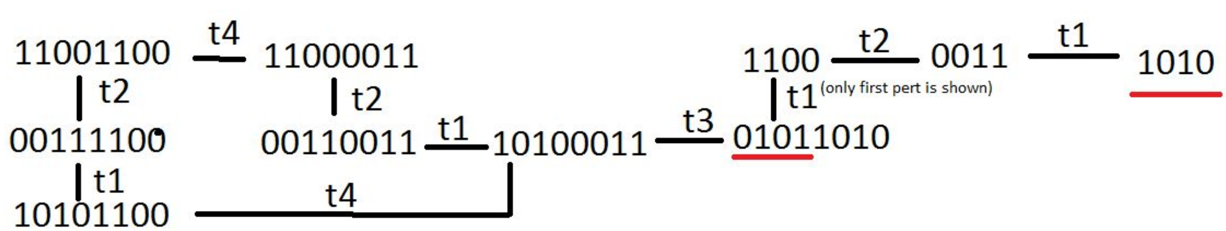
**1.** 1. Let's build a reachability graph for first Petri net. In representation, we will use binary values to show places (Zero means there is no token in the place, and one means there is one, and the index means the place number)




From the reachability graph, it is seen that it needs 2 transitions to fire, in order to get all odd numbered places to have a token.

$Mf_1 = 2$

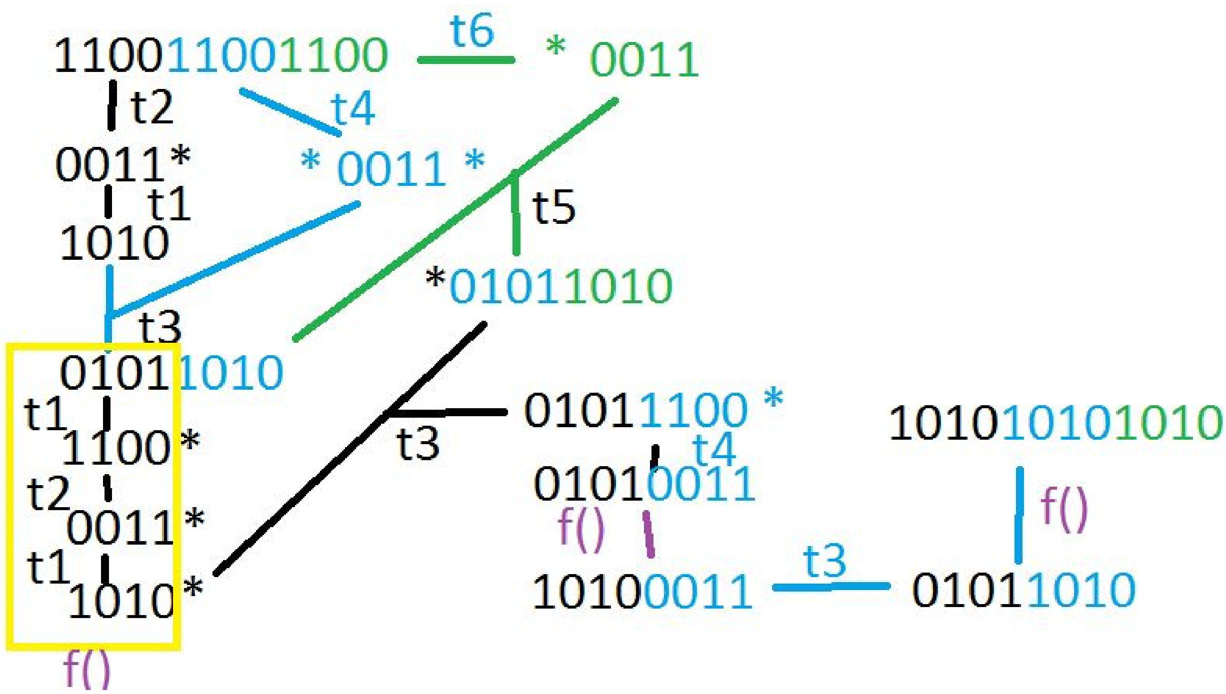
2. Now we build a reachability for Petri net 2.




We can start firing transitions in different ways, but at the end they converge into the 10100011 marking. And then we arrive at 01011010. After that, the right part of Petri net is set, and the only thing is possible is to make left part right.

$Mf_2 = 7$

3. While we build the reachability graph for the N = 3, we start to see some patterns.




The f() is the repeatable pattern that has 3 transitions fired.

The counting the number of transitions fired yields us the number of 9+3*3= 18.

$Mf_3 = 18$

## 2.

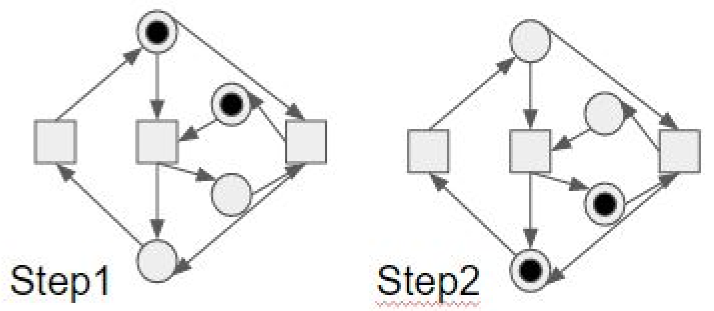
$Mf_n = 3 \cdot (2^n - 1) - n$ (formula 1)

**Proof.**

The proof is by induction by the number of Petri net parts.

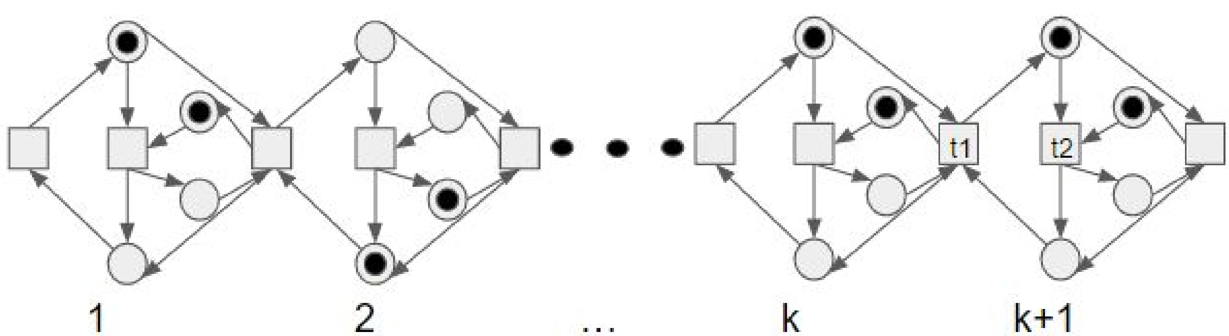
The base case for Petri net $P$ with $n = 1$ is $Mf_1 = 3 \cdot (2^1 - 1) - 1 = 2$ number of minimum transitions.

This can be shown by firing two transitions

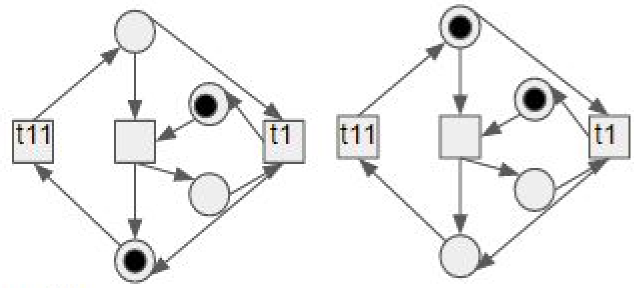



Now, suppose that Petri net $P$ has $n = k$ parts. We assume that for $P$ with $k$ parts the minimum number of firings is $Mf_k = 3 \cdot (2^k - 1) - k$. We must prove that the formula is true for $n = k + 1$.

Let's assume that we have a petri net with $k$ parts. We add one new part to the left of the sequence.




In order to move tokens in left most tree to the odd numbered places, we need minimum two firings (as for the base case). The minimum sequence is to fire first $t2$ and then $t1$ transition. In order to fire $t1$ it should be enabled by tokens being in odd numbered places from $k$ part. After $t1$ fires tokens in part $k$ are even numbered places. To "fix" the $k$ part, you should first by firing the transition $t11$ achieve initial marking.




part k
after firing t1

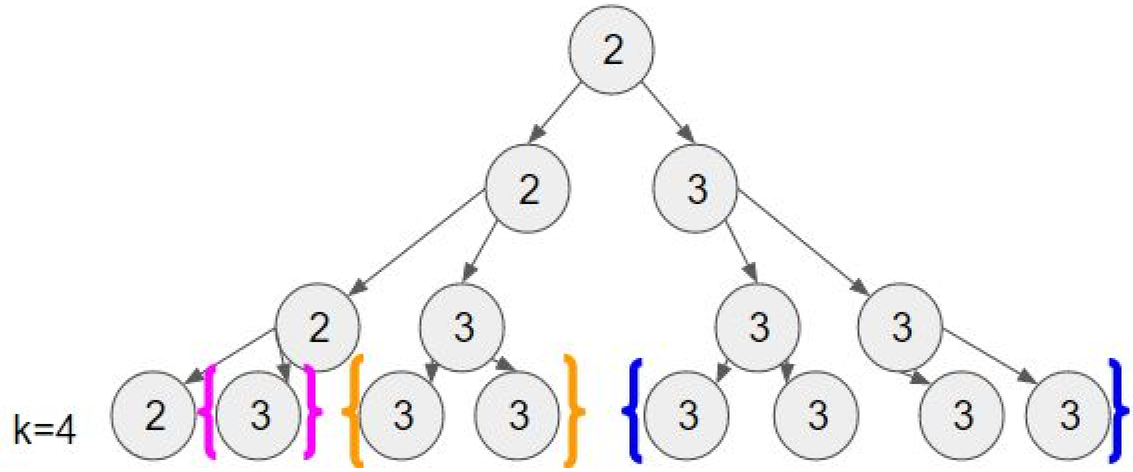
part k
after firing t11
in its initial markings

That means that after the $k$ part's marking rearranged after firing $t1$ it costs additional one firing (in comparison with initial marking) to get to odd numbered places. It involves recursive firings in all leftside parts.

$Mf_k = 3 \cdot (2^{k+1} - 1) - (k + 1)$

By adding one new part, we add two firings to fix the rightmost part and $2^0 \cdot 3$ to fix part $k$. That proves the formula.

To understand formula better it can be graphed as a binary tree. The root of a tree is the number of firings needed to get the rightmost part of a tree fixed




For $k = 1$ , it is two firing needed to get the right result.
For $k = 2$ , it is needed two for the rightmost part, and 2 + 3 for the left part (2 - to get odd numbered from initial marking, and plus tree after firing state common with rightmost part)
It is evident that every new part adds new layer of nodes to binary tree. It adds 2 firings as a new part, and we can calculate how many firings are added to any part by adding a new layer by formula: $newparts_i = 3 \cdot 2^i$ (where $i = partnumber - 1$ )

The number of nodes in the full binary tree we can calculate by this approach: The first layer, all consecutive have the form:
1,2,4,8,16.. so we can calculate a sum of geometric progression, that gives us:
$nodes_i = \frac{1-2^n}{1-2} = \frac{1-2^n}{-1} = 2^n - 1$
As in every except one node every layer we have 3 firings we can get the formula:
$Mf_n = 3 \cdot (2^n - 1) - n$