

The writers of the subject article, “A Modified Reachability Tree Approach to Analysis of Unbounded Petri Nets” (hereafter simply called “the article”), contend that the current method of designing coverability trees for Petri nets (what are referred to in the article as “Finite Reachability Trees,” or FRT) creates an unnecessary loss of information. Certain facts about the Petri net described by the FRT are made indistinct by the use of the “ ω ” symbol to indicate an unbounded place.

For example, consider a 3-place Petri net set to the state $[1\ 0\ 1]^T$. Suppose a transition T1 only adds 1 token to place P1 when fired. The result will be $[\omega\ 0\ 1]^T$, as shown in Figure 1. However, if T1 added 2 tokens to P1, it would have the same result, making it impossible to accurately reverse-engineer the Petri net from its FRT.

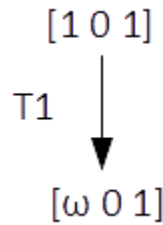


Figure 1: Transition involving “ ω ”

The article proposes the development of a “Modified Reachability Tree” (MRT), which will introduce additional operations to the FRT construction process. These operations will cause the effects on the place values replaced by “ ω ” to be visible, allowing the user to, in theory, use the MRT to determine the Liveness, Reachability, and Deadlock status of a Petri net, as well as determine the firing sequence of unbounded Petri nets.

This review will examine the exact process put forth in the article for the construction of MRTs, as well as its effectiveness in addressing the issues given above. Further improvements that could be made to the MRT process will also be discussed.

The best way to begin is by discussing the actual process for constructing MRTs, as proposed in the article. Obviously, construction of the coverability tree proceeds normally up to the point that an ω would be used. At that point, the circumstances involved in creating the unbounded state must be taken into account (as is the purpose of the Modified Reachability Tree). First, the ω is given a subscript to indicate how many times the same operation has been conducted. Thus, the first time an ω occurs, it will be labeled “ ω_1 ”. If the same transition happens again, it will result in “ ω_2 ”. If an operation occurred to remove the same number of tokens from the ω state as were originally transferred to it, the subscript would decrease (for instance, from ω_1 to ω_0).

The next part of the MRT conversion introduces a numerical coefficient to the ω to indicate the number of tokens added to the place to bring it to its ω state. For example, if a transition adds two tokens to the target place, P1, the result will be $2\omega_1$. If the transition adds 3 tokens, it will generate a state of $3\omega_1$ in P1. Every time the same transition fires, (or, for that matter, any transition depositing the same number of tokens in P1), the subscript will advance, such that the state will have an appearance similar to “ $2\omega_2$ ”. In other words, as long as transitions add to- or subtract from the ω -value place in multiples of the numerical coefficient,

the subscript numbers behave as has already been specified.

Suppose, however, that a different transition fires, which adds a different number of tokens. The Modified Reachability Tree approach handles such situations with a “+x” construct, where “x” is any non-negative integer number. Return to the example mentioned earlier, where P1 has a state of $3\omega_1$. If a transition fires that adds a single token to P1, the state would be given as $3\omega_1+1$. Alternatively, if a transition removes one token from P1, the state would then be $3\omega_0+2$. If the ω has a coefficient of 1, there will be no reason to display an “x”.

At this point, the question of how negative numbers are handled might come to mind. If a transition would cause a place that already contains an ω state to drop below 0 tokens, that transition is indicated with a dotted line. However, the ω subscript will itself never drop below 0. Once again, examine the earlier “ $3\omega_1$ ” example. If a transition T2 removes 3 tokens from place P1, the resulting state will be $3\omega_0$. If T2 fires again, the state will still read as “ $3\omega_0$ ”, however, the transition will be represented graphically as a dotted line. If a transition were to fire that subtracted less than 3 tokens from $3\omega_0$, the result would be of the form $3\omega_0+x$, where “x” is the coefficient, 3, minus the number of tokens subtracted. For example, if, from a state of $3\omega_0$, 2 tokens were removed, the resulting state would be $3\omega_0+1$.

This entire process is unintuitive and difficult to explain with words alone, especially considering that the Modified Reachability Tree is graphical in nature anyway. Figure 2 below shows an example Petri net, taken from the article. Figure 3 shows the Petri net’s FRT, for comparison. Figure 4 shows the MRT for the Petri net. This particular example will demonstrate all of the elements of the MRT in use.

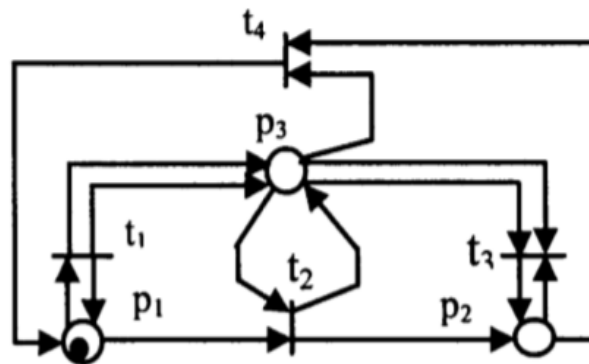


Figure 2: An example Petri net

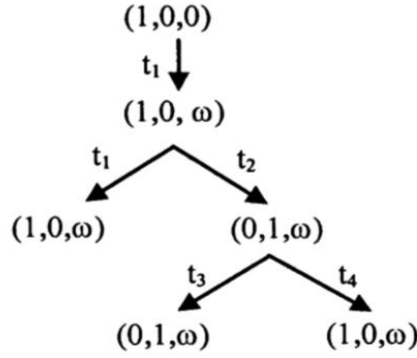


Figure 3: The standard Finite Reachability Tree for Figure 2

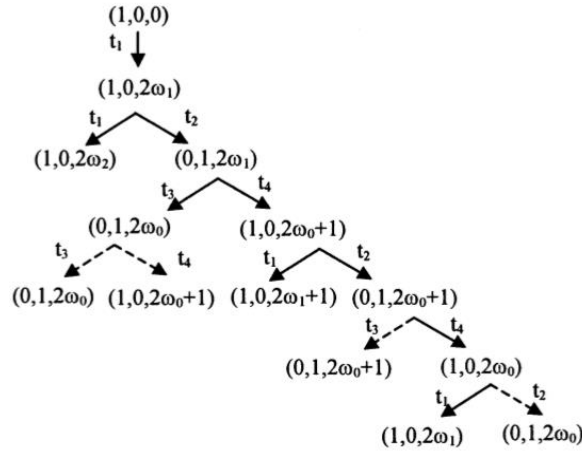


Figure 4: The Modified Reachability Tree for Figure 2

One interesting point to note is that the MRT can be much longer than its equivalent FRT. This arises simply from the fact that many ω terms are no longer duplicates of each other.

At this point, the actual review of the process should be discussed. The reviewers are of the firm belief that the Modified Reachability Tree is far more complex than necessary. Although part of the review is supposed to involve the reviewers' own additions to the concepts presented in the subject article, we hold that the best improvement we could make would be to take some of the MRT away. As it stands, the MRT provides too much information, some of it redundant, making the process cumbersome, and difficult for newcomers to understand, even when they have had experience with FRT systems in the past.

Additionally, any time serious modifications to a system like FRT are made, complications can arise. One such complication arises from the MRT's definition of deadlock states. MRT Theorem 2 states, "The modified reachability tree of a Petri net consists of only but all reachable markings from its initial marking." Although grammatically confusing, Theorem 2 suggests that an MRT diagram consists of only reachable states. This suggests that any state resulting from a conditional transition is not actually part of the MRT. However, MRT Theorem 3 states that, "A Petri net has deadlocks if and only if its modified reachability tree contains terminal nodes and/or full conditional nodes." In this case, a full conditional node refers to a node which has only conditional transitions, as shown in Figure 5. According to Yu Ru, Weimin

Wu, and Christoforos N. Hadjicostis in, “Comments on ‘A Modified Reachability Tree Approach to Analysis of Unbounded Petri Nets,’” Theorem 3 can come into conflict with MRT Theorem 2 under certain circumstances.

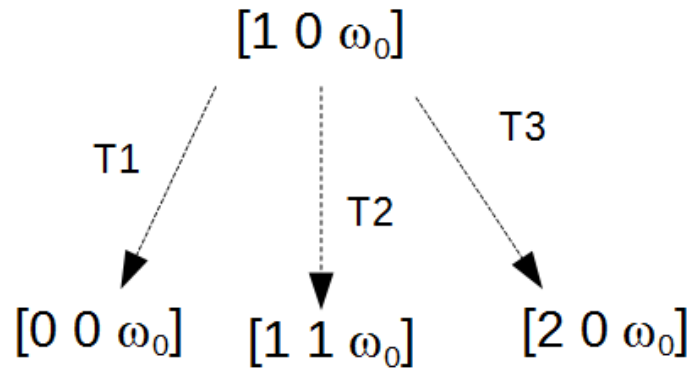


Figure 5: A Full-Conditional node in MRT

For example, consider the Petri net in Figure 6, and its MRT in Figure 7. According to Theorem 3, the state reached by the transition sequence t_2, t_1, t_3, t_1 should be considered a deadlock state. However, Theorem 2 suggests that the state is not an official part of the MRT, since it is unreachable, and thus cannot be a deadlock state.

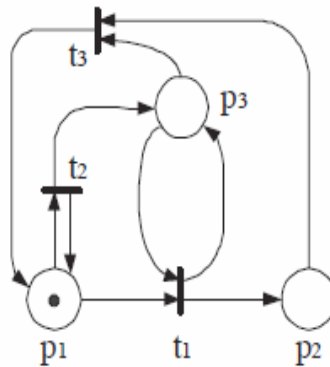


Figure 6: A Petri net with no apparent deadlock states (courtesy of [2])

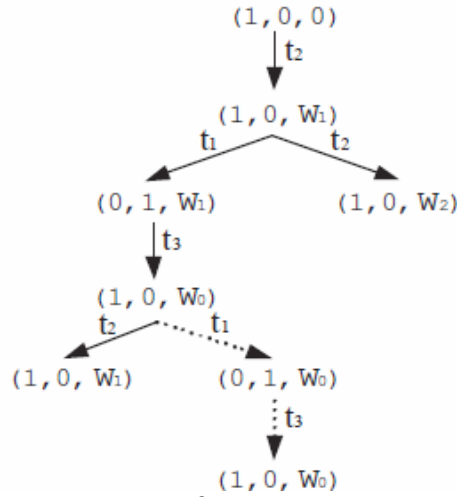


Figure 7: MRT of Figure 6 (courtesy of [2])

A few solutions have been proposed to solve this deadlock paradox, and two in particular stand out. The first solution, proposed by ZhiJun Ding, ChangJun Jiang, and MengChu Zhou in the article “Deadlock Checking for One-Place Unbounded Petri Nets Based on Modified Reachability Trees”, involves the creation of an integer variable, “r”. If a state is located that fulfills conditions to be considered in deadlock, but is itself the result of a conditional transition, the branch of the MRT is traced backwards to locate the first occurrence of an ω value. Initially, r is set to 0, and added to that first ω value. If the transition in question remains conditional after the new ω value propagates through the MRT, then the value of r is incremented by 1, and the process is attempted again. Once a value is found for r that allows the transition to fire without being considered conditional, then the resulting state is checked for deadlock conditions again. If the state still has only conditional transitions, it is considered a true conditional node. Otherwise, if there is at least one transition branching from the formerly-deadlocked state that is not conditional due to the new ω value, the state is considered a fake conditional node.

The other solution involves modifying the MRT further, into the New Modified Reachability Tree, or NMRT. This method was proposed by ShouGuang Wang, MengChu Zhou, ZhiWu Li, and ChengYing Wang in “A New Modified Reachability Tree Approach and Its Applications to Unbounded Petri Nets”. The NMRT method takes advantage of the fact that an ω value in a place is operationally infinite, and thus embodies an infinite set of integer values with the given numbered value for ω set as the least bound. The means of handling a conditional deadlock state is similar to the previous method, but instead of assigning the variable “r” a value and adding it to the ω subscript, the ω place is simply advanced to the next possible integer in the set. The process is functionally similar to the previous method, except that the explanation takes into account the effect of the +x and coefficient elements of the MRT more clearly. Either way, the same effect occurs, which is that the ω value is advanced until the transition immediately before the deadlock state becomes non-conditional, then the deadlock state is treated accordingly, using the newly-propagated value for ω .

With the deadlock problem, and its solutions, in mind, the reviewers have designed their own FRT revision. The process to obtain this new reachability tree is given as follows.

It is axiomatic to Petri nets that only one transition may fire at a time. That is just a given rule of how Petri nets are processed. Thus, when a transition occurs, it can be safely assumed that all differences that occurred between the beginning state and the end state must have been caused by that one transition. Therefore, having ω states of the form “ $y\omega_z+x$ ” is unnecessary. There is no reason to specifically label an ω state caused by adding two tokens as “ $2\omega_1$ ” when the 2 could be placed in the subscript. The same logic pertains to the “ $+x$ ” element, which really does not need to be used while the number of tokens associated with the ω state can be displayed fully in the subscript.

On the other hand, the dotted-line notation, used for dealing with potential negative token values, should be kept intact. Using a dotted line to indicate transitions that are only possible when an infinite value is assumed for a place containing an ω state may make it more clear when a transition will not be possible in the real-world execution of a Petri net (at least, without additional intermediate steps). Negative numbers for the subscript may be possible, but ultimately indicate only how many additional intermediate steps will be necessary (by indicating the token shortage created by the transition).

Additionally, for this new approach, all ω states should be considered duplicates if the later-occurring ω state has a larger subscript value associated with the ω place, otherwise the tree may continue on without end, defeating the purpose of a *Finite* Reachability Tree. In other words, if the state $[0 \ \omega_2 \ 1]^T$ exists in an FRT, and sometime later, the state $[0 \ \omega_3 \ 1]^T$ occurs, the two states should be considered duplicates, even though the token value given in the subscript is different.

To avoid any future confusion, we will refer to this new system as the Lindley-Al Otaibi Tree, or LAT. The rules for the LAT are summarized as follows:

1. All theoretical token values for unbounded places are presented as a subscript “z” for ω , in the form “ ω_z ”.
2. An unbounded node is considered duplicate if it contains markings identical to a previous state, with an ω value equal to or greater than that of the previous state.
3. Conditional nodes, represented by a dotted/dashed line transition, are states that may only be accessed from the parent state if an infinite value is assumed for an ω state.
4. The ω subscript for such states will be a negative number to show the token deficit created by the conditional transition.
5. An unbounded state is considered terminal if it is fully conditional, and duplicate of a previous state containing a larger ω value.
 - a. Optionally, if such a state originates from a conditional transition, it is considered a fake terminal node.
6. An unbounded state is once again considered duplicate if it results from a conditional transition, is not fully-conditional itself, and is identical to a previous state, with an ω value less than that of the previous state.

Now that a distinction has been drawn between the Modified Reachability Tree model and the Lindley-Al Otaibi Tree model, the question must invariably be raised: does the LAT solve the problems presented by the MRT? Simplicity has already been covered, and the LAT is

demonstrably simpler to construct. As for the deadlock issue, the LAT has been tested against the Petri net of Figure 6, and found (when point 5a is applied) to identify the fake conditional node correctly. Point 5a is given as optional simply because the deadlock issue may not be considered a problem for some users of the LAT. As the token deficit is given in the ω subscript, it is made clear what conditions should be met to enable any conditional transitions.

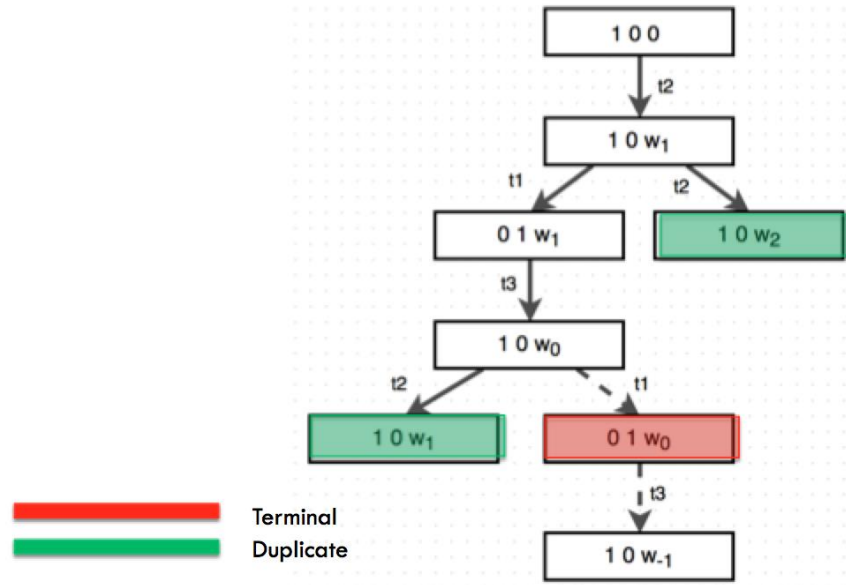


Figure 8: The LAT of the Figure 6 Petri net

In addition, the LAT for the Petri net of Figure 2 is given in Figure 9 for comparison. Note that the state given by the transition sequence t_1, t_2, t_3 is a deadlock state, and the next two states are given to show that it is fully conditional.

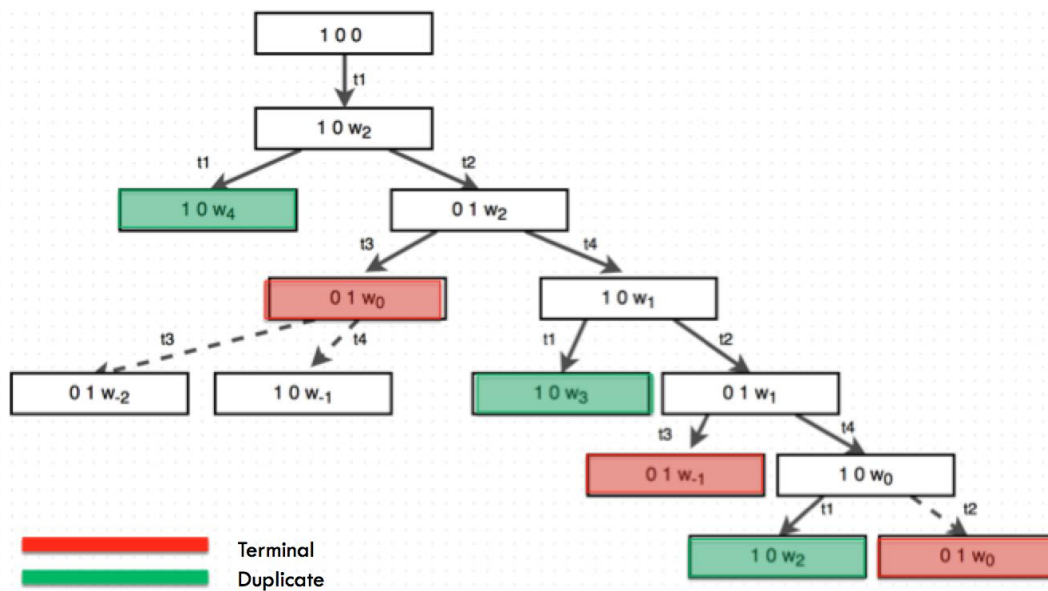


Figure 9: the LAT of the Figure 2 Petri net

Figure 10 shows yet another Petri net and its associated LAT, as an example of why point 6 was added to the LAT method. While point 2 considered the fact that a coverability tree could continue without end if a transition simply added more tokens to an ω value, point 6 considers that the same can happen for transitions that remove tokens from an ω value. The token deficit could simply become more and more negative, without any conditions to either declare a terminal state or a duplicate state.

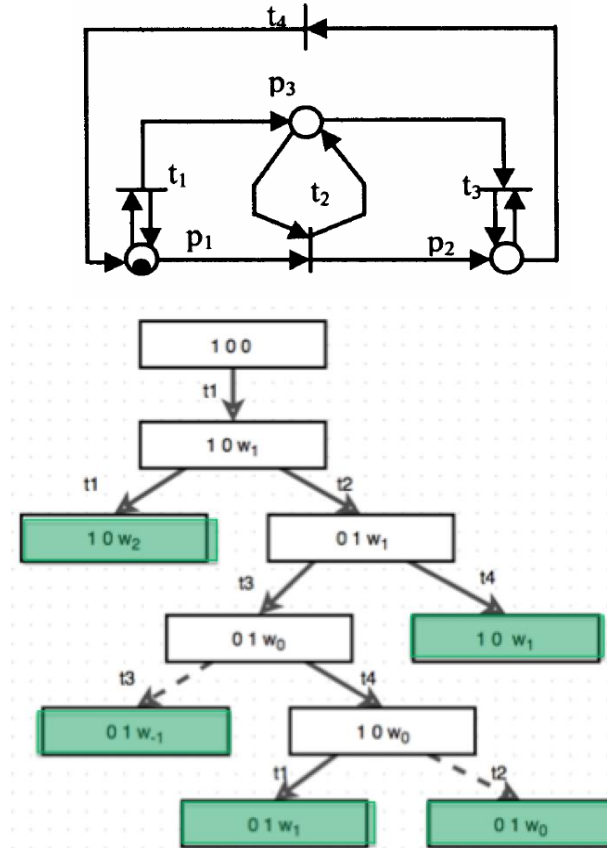


Figure 10: A third Petri net and its LAT

The MRT method provides a good basis for reintroducing information about the subject Petri net into a Finite Reachability Tree. However, the MRT process created an issue by simultaneously declaring unreachable states to be outside of the MRT, and determining all fully-conditional states to be deadlock nodes, which allowed deadlock nodes to exist outside of the tree in certain cases. Both the deadlock checking process and the NMRT approach provide workable solutions to the apparent contradiction of unreachable deadlock states. Finally, the Lindley-Al Otaibi Tree method was developed to create a simplified reachability tree format that also solves the uncertainty created by the unreachable deadlock problem.

References:

1. “A Modified Reachability Tree Approach to Analysis of Unbounded Petri Nets”, Fei-Yue Wang, Yanqing Gao, and MengChu Zhou, Fellow, IEEE
2. “Comments on ‘A Modified Reachability Tree Approach to Analysis of Unbounded Petri Nets’”, Yu Ru, Weimin Wu, Christoforos N. Hadjicostis
3. “Deadlock Checking for One-Place Unbounded Petri Nets Based on Modified Reachability Trees”, ZhiJun Ding, ChangJun Jiang, and MengChu Zhou, Fellow, IEEE
4. “A New Modified Reachability Tree Approach and Its Applications to Unbounded Petri Nets”, ShouGuang Wang, Senior Member, IEEE, MengChu Zhou, Fellow, IEEE, ZhiWu Li, Senior Member, IEEE, and ChengYing Wang