

# Towards Effective Elicitation of NIN-AND Tree Causal Models

Yang Xiang, Yu Li, and Zoe Jingyu Zhu

University of Guelph, Canada

**Abstract.** To specify a Bayes net (BN), a conditional probability table (CPT), often of an effect conditioned on its  $n$  causes, needs assessed for each node. It generally has the complexity exponential on  $n$ . Noisy-OR reduces the complexity to linear, but can only represent reinforcing causal interactions. The non-impeding noisy-AND (NIN-AND) tree is the first causal model that explicitly expresses reinforcement, undermining, and their mixture. It has linear complexity, but requires elicitation of a tree topology for types of causal interactions. We study their topology space and develop two novel techniques for more effective elicitation.

## 1 Introduction

To specify a BN, a CPT needs to be assessed for each non-root node. It is often advantageous to construct BNs along the causal direction, in which case a CPT is the distribution of an effect conditioned on its  $n$  causes. In general, assessment of a CPT has the complexity exponential on  $n$ .

Noisy-OR [7] is the most well known technique that reduces this complexity to linear. A number of extensions have also been proposed such as [4, 3, 5]. However, noisy-OR, noisy-AND [3], as well as related techniques, can only represent causal interactions that are reinforcing [9]. The NIN-AND tree [9] extends noisy-OR and provides the first causal model that explicitly expresses reinforcing and undermining causal interactions, as well as their mixture. It requires elicitation of a linear number of probability parameters and, in addition, a tree topology which specifies the types of causal interactions among causes.

The elicitation relies on expert to describe the tree topology. When the number of causes is more than four or five, accurate description may be challenging. We study the topology space of NIN-AND tree models and develop novel techniques for more effective elicitation. One allows expert to select a topology from an enumeration. Another allows expert to specify only types of pairwise interactions among causes, from which a unique tree topology is identified.

## 2 Background

This section is mostly based on [9]. An *uncertain cause* is a cause that can produce an effect but does not always do so. Denote a set of binary cause variables as  $X = \{c_1, \dots, c_n\}$  and their effect variable (binary) as  $e$ . For each  $c_i$ , denote

$c_i = \text{true}$  by  $c_i^+$  and  $c_i = \text{false}$  by  $c_i^-$ . Similarly, denote  $e = \text{true}$  by  $e^+$  and  $e = \text{false}$  by  $e^-$ .

A *causal event* refers to an event that a cause  $c_i$  caused an effect  $e$  to occur successfully when all other causes of  $e$  are absent. Denote this causal event by  $e^+ \leftarrow c_i^+$  and its probability by  $P(e^+ \leftarrow c_i^+)$ . The causal failure event, where  $e$  is false when  $c_i$  is true and all other causes of  $e$  are false, is denoted as  $e^+ \not\leftarrow c_i^+$ . Denote the causal event that a set  $X = \{c_1, \dots, c_n\}$  of causes caused  $e$  by  $e^+ \leftarrow c_1^+, \dots, c_n^+$  or  $e^+ \leftarrow \underline{x}^+$ . Denote the set of *all causes* of  $e$  by  $C$ . The CPT  $P(e|C)$  relates to probabilities of causal events as follows: If  $C = \{c_1, c_2, c_3\}$ , then  $P(e^+|c_1^+, c_2^-, c_3^+) = P(e^+ \leftarrow c_1^+, c_3^+)$ . Note that  $C$  is assumed to include a leaky variable (if any) to capture causes that we do not wish to represent explicitly, and hence  $P(e^+|c_1^-, c_2^-, c_3^-) = 0$ .

Causes reinforce each other if collectively they are at least as effective in causing the effect as some acting by themselves. If collectively they are less effective, then they undermine each other. Note that if  $C = \{c_1, c_2\}$  and  $c_1$  and  $c_2$  undermine each other, then all the following hold:

$$P(e^+|c_1^-, c_2^-) = 0, \quad P(e^+|c_1^+, c_2^-) > 0, \quad P(e^+|c_1^-, c_2^+) > 0,$$

$$P(e^+|c_1^+, c_2^+) < \min(P(e^+|c_1^+, c_2^-), P(e^+|c_1^-, c_2^+)).$$

The following Def.1 defines the two types of causal interactions generally. Note that reinforcement and undermining occur between individual variables as well as sets of variables. For instance, variables within each of two sets can be reinforcing, while the two sets can undermine each other. Hence, each  $W_i$  in Def.1 is not necessarily a singleton.

**Def. 1** Let  $R = \{W_1, W_2, \dots\}$  be a partition of a set  $X$  of causes,  $R' \subset R$  be any proper subset of  $R$ , and  $Y = \cup_{W_i \in R'} W_i$ . Sets of causes in  $R$  reinforce each other, iff

$$\forall R' P(e^+ \leftarrow \underline{y}^+) \leq P(e^+ \leftarrow \underline{x}^+).$$

Sets of causes in  $R$  undermine each other, iff

$$\forall R' P(e^+ \leftarrow \underline{y}^+) > P(e^+ \leftarrow \underline{x}^+).$$

Disjoint sets of causes  $W_1, \dots, W_m$  satisfy *failure conjunction* iff

$$(e^+ \not\leftarrow \underline{w}_1^+, \dots, \underline{w}_m^+) = (e^+ \not\leftarrow \underline{w}_1^+) \wedge \dots \wedge (e^+ \not\leftarrow \underline{w}_m^+).$$

That is, collective failure is attributed to individual failures. They also satisfy *failure independence* iff

$$P((e^+ \not\leftarrow \underline{w}_1^+) \wedge \dots \wedge (e^+ \not\leftarrow \underline{w}_m^+)) = P(e^+ \not\leftarrow \underline{w}_1^+) \dots P(e^+ \not\leftarrow \underline{w}_m^+).$$

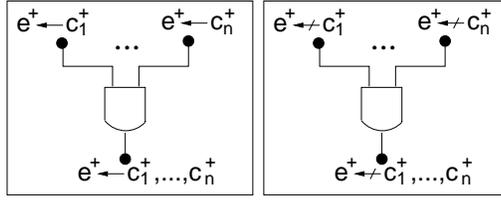
Disjoint sets of causes  $W_1, \dots, W_m$  satisfy *success conjunction* iff

$$e^+ \leftarrow \underline{w}_1^+, \dots, \underline{w}_m^+ = (e^+ \leftarrow \underline{w}_1^+) \wedge \dots \wedge (e^+ \leftarrow \underline{w}_m^+).$$

That is, collective success requires individual effectiveness. They also satisfy *success independence* iff

$$P((e^+ \leftarrow \underline{w}_1^+) \wedge \dots \wedge (e^+ \leftarrow \underline{w}_m^+)) = P(e^+ \leftarrow \underline{w}_1^+) \dots P(e^+ \leftarrow \underline{w}_m^+).$$

It has been shown that causes are reinforcing when they satisfy failure conjunction and independence, and they are undermining when they satisfy success conjunction and independence. Hence, undermining can be modeled by a direct



**Fig. 1.** Direct (left) and dual (right) NIN-AND gates

NIN-AND gate (Fig. 1, left), and reinforcement by a dual NIN-AND gate (right).

As per Def. 1, a set of causes can be reinforcing (undermining), but the set is undermining (reinforcing) with another set. Such causal interaction can be modeled by a NIN-AND tree. As shown in Fig. 2 (a), causes  $c_1$  through  $c_3$  are undermining, and they are collectively reinforcing  $c_4$ . The following defines NIN-AND tree models in general:

**Def. 2** An NIN-AND tree is a directed tree for effect  $e$  and a set  $X = \{c_1, \dots, c_n\}$  of occurring causes.

1. There are two types of nodes. An **event node** (a black oval) has an in-degree  $\leq 1$  and an out-degree  $\leq 1$ . A **gate node** (a NIN-AND gate) has an in-degree  $\geq 2$  and an out-degree 1.
2. There are two types of links, each connecting an event and a gate along input-to-output direction of gates. A **forward link** (a line) is implicitly directed. A **negation link** (with a white oval at one end) is explicitly directed.
3. Each terminal node is an event labeled by a causal event  $e^+ \leftarrow \underline{y}^+$  or  $e^+ \not\leftarrow \underline{y}^+$ . There is a single **leaf** (no child) with  $\underline{y}^+ = \underline{x}^+$ , and the gate it connects to is the **leaf gate**. For each **root** (no parent; indexed by  $i$ ),  $\underline{y}_i^+ \subset \underline{x}^+$ ,  $\underline{y}_j^+ \cap \underline{y}_k^+ = \emptyset$  for  $j \neq k$ , and  $\bigcup_i \underline{y}_i^+ = \underline{x}^+$ .
4. Inputs to a gate  $g$  are in one of two cases:
  - (a) Each is either connected by a forward link to a node labeled  $e^+ \leftarrow \underline{y}^+$ , or by a negation link to a node labeled  $e^+ \not\leftarrow \underline{y}^+$ . The output of  $g$  is connected by a forward link to a node labeled  $e^+ \leftarrow \bigcup_i \underline{y}_i^+$ .
  - (b) Each is either connected by a forward link to a node labeled  $e^+ \not\leftarrow \underline{y}^+$ , or by a negation link to a node labeled  $e^+ \leftarrow \underline{y}^+$ . The output of  $g$  is connected by a forward link to a node labeled  $e^+ \not\leftarrow \bigcup_i \underline{y}_i^+$ .

An NIN-AND tree model for effect  $e$  and its causes  $C$  can be obtained by eliciting a tree topology with  $|C|$  roots plus  $|C|$  single-cause probabilities  $P(e^+ \leftarrow c_i^+)$ . The CPT  $P(e|C)$  can then be derived using the model. For each  $P(e^+|c_i, \dots, c_j)$ , first modify the model to remove roots corresponding to inactive causes, i.e.,  $c_i = c_i^-$ , and related gates if necessary. Then apply algorithm `GetCausalEventProb` below to the modified tree. It recursively processes from the leaf to roots. As soon as probabilities of input events to a gate is obtained, probability of its output event is computed.

**Algorithm 1** *GetCausalEventProb(T)*

*Input: A NIN-AND tree T of leaf v and leaf gate g, with root probabilities specified.*

*for each node w directly inputting to g, do*  
*if P(w) is not specified,*  
*denote the sub-NIN-AND-tree with w as the leaf by T<sub>w</sub>;*  
*P(w) = GetCausalEventProb(T<sub>w</sub>);*  
*if (w, g) is a forward link, P'(w) = P(w);*  
*else P'(w) = 1 - P(w);*  
*return P(v) = ∏<sub>w</sub> P'(w);*

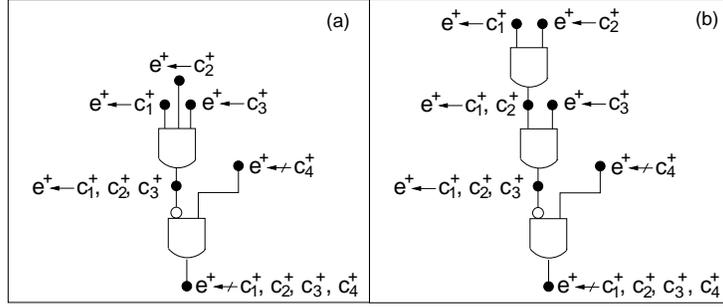
By default, each root event in a NIN-AND tree is a single-cause event, and all causal interactions satisfy failure (or success) conjunction and independence. If a subset of causes do not satisfy these assumptions, suitable multi-cause probabilities  $P(e^+ \leftarrow \underline{x}^+)$ , where  $X \subset C$ , can be directly elicited and incorporated into the NIN-AND tree model. Hence, by trading efficiency, any non-deterministic CPT can be encoded through NIN-AND trees. The default is assumed in this paper.

### 3 Minimal NIN-AND Tree Topology Space

NIN-AND tree models allow a CPT of generally exponential complexity to be obtained by eliciting a tree topology and a linear number of probabilities of single-cause events. Reference [9] relies on human expert to describe the tree topology. One alternative is to show expert all possible tree topologies so that one can be selected. We study the space of NIN-AND trees below so that tree topologies can be enumerated.

First of all, what qualifies as an individual in the space? For instance, it would be undesirable that two distinct topologies in the space correspond to the same CPT. Consider the two NIN-AND trees in Fig. 2. Although the topologies appear different, given an identical set of single-cause probabilities, they yield the same probability  $P(e^+ \leftarrow c_1^+, c_2^+, c_3^+, c_4^+)$ . Hence, it is desirable that only one of them is deemed legal in the topology space.

We establish below associativity of NIN-AND gates, which allows identification of equivalent topologies such as the above. Proposition 1 shows that an NIN-AND tree of multiple dual NIN-AND gates is equivalent to a single gate.



**Fig. 2.** NIN-AND trees depicting the same causal model.

**Proposition 1** *Let  $T$  be an NIN-AND tree for  $n \geq 3$  causes with  $m \geq 1$  dual NIN-AND gates, and  $P(e^+ \leftarrow c_1^+, \dots, c_n^+)$  be the probability of the causal event obtained from  $T$ . Let  $T'$  be a single dual NIN-AND gate for the same causes and  $P'(e^+ \leftarrow c_1^+, \dots, c_n^+)$  the probability from  $T'$ . Then  $P(e^+ \leftarrow c_1^+, \dots, c_n^+) = P'(e^+ \leftarrow c_1^+, \dots, c_n^+)$ .*

Proof: We prove by induction on the number of gates in  $T$ . For  $m = 1$ ,  $T$  and  $T'$  are identical. Applying GetCausalEventProb to  $T$ , we obtain

$$P(e^+ \leftarrow c_1^+, \dots, c_n^+) = P'(e^+ \leftarrow c_1^+, \dots, c_n^+) = \prod_{i=1}^n P(e^+ \leftarrow c_i^+).$$

Assume that the proposition holds for  $m = 1, 2, \dots, k$  where  $k \geq 1$ . Below we consider  $m = k + 1$ .

Let  $g$  denote the leaf gate of  $T$ . Since  $k \geq 1$  and  $m \geq 2$ , there exists a gate  $t$  that outputs to  $g$ . Let  $v$  be the output event of  $t$ . Let  $S$  denote the subtree seated at  $v$  and for a subset  $X$  of causes. In other words,  $v$  is the leaf in  $S$ .  $S$  is a valid NIN-AND tree with no more than  $k$  gates. By inductive assumption, applying GetCausalEventProb to  $S$ , we obtain

$$P(e^+ \leftarrow \underline{x}^+) = \prod_{c_i \in X} P(e^+ \leftarrow c_i^+).$$

The above argument holds for each gate  $t$ . When GetCausalEventProb is applied to the leaf node in  $T$ , the probability of the corresponding event is the product of the probability of each input event to  $g$ . Hence,

$$P(e^+ \leftarrow c_1^+, \dots, c_n^+) = \prod_{i=1}^n P(e^+ \leftarrow c_i^+)$$

when  $m = k + 1$ .  $\square$

Proposition 2 shows that an NIN-AND tree of multiple direct NIN-AND gates is equivalent to a single direct NIN-AND gate. It can be proven similarly as for Proposition 1.

**Proposition 2** *Let  $T$  be an NIN-AND tree for  $n \geq 3$  causes with  $m \geq 1$  direct NIN-AND gates, and  $P(e^+ \leftarrow c_1^+, \dots, c_n^+)$  be the probability of the causal event obtained from  $T$ . Let  $T'$  be an NIN-AND tree for the same causes with a single direct NIN-AND gate, and  $P'(e^+ \leftarrow c_1^+, \dots, c_n^+)$  be the probability from  $T'$ . Then  $P(e^+ \leftarrow c_1^+, \dots, c_n^+) = P'(e^+ \leftarrow c_1^+, \dots, c_n^+)$ .*

Base on the associativity of NIN-AND gates, we select the single NIN-AND gate to represent all equivalent NIN-AND trees of the same input events, and to be the only legal individual in the topology space. Applying this to NIN-AND gates embedded in an NIN-AND tree, we have the following classification of tree topologies.

**Def. 3** *Let  $T$  be an NIN-AND tree. If  $T$  contains a gate  $t$  that outputs to another gate  $g$  of the same type (direct or dual), delete  $t$  and connect its inputs to  $g$ . If such deletion is possible, then  $T$  is **superfluous**. Apply such deletions until no longer possible. The resultant NIN-AND tree is **minimal**.*

As per Def. 3, we require individuals in the topology space to be minimal. That is, we require a *minimal* topology space. Hence, the NIN-AND tree in Fig. 2 (a) is legal in the space, and that in (b) is not. This leads to Corollary 1.

**Corollary 1** *Let  $T$  be a minimal NIN-AND tree. Then whenever a NIN-AND gate  $g$  outputs to another NIN-AND gate  $t$ ,  $g$  and  $t$  are of different type (direct or dual).*

From Corollary 1, a minimal NIN-AND tree has the following structure: If the leaf gate  $g$  is a direct gate, then all gates outputting to  $g$  are dual, and their inputs are all from direct gates. That is, from the leaf towards root nodes, gates alternate in types. This alternation implies that, in the minimal space, for every legal NIN-AND tree  $T$  with a direct leaf gate, there exists a legal NIN-AND tree  $T'$  obtained by replacing each gate in  $T$  with its opposite type. This is summarized in the following:

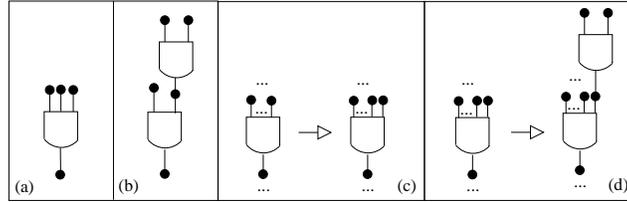
**Proposition 3** *Let  $\Psi$  be the collection of minimal NIN-AND trees for  $n$  causes with direct leaf gates. Let  $\Psi'$  be the collection of minimal NIN-AND trees for  $n$  causes with dual leaf gates. Then an one-to-one mapping exists between  $\Psi$  and  $\Psi'$ , defined by replacing each gate with the opposite type.*

We refer to  $\Psi$  as the minimal topology space for  $n$  causes with direct leaf gates, and  $\Psi'$  as the minimal topology space with dual leaf gates. As per Proposition 3, properties from one of them are applicable to the other. Below, we focus on tree enumeration in the minimal space with direct leaf gates.

From Corollary 1, given the type of leaf gate, types of all gates in a minimal NIN-AND tree are unique, as well as the nature of all event nodes (causal failure or success). We thus omit labels for event nodes. Note that we choose minimal trees to be *unlabeled* as the space is more compact. How to enumerate root-labeled minimal trees given the unlabeled enumeration is studied in [10].

## 4 Enumerating NIN-AND Trees by Local Insertion

There is only one minimal NIN-AND tree for 2 causes. For 3 causes, there are two minimal NIN-AND trees with direct leaf gate (see Fig. 3 (a) and (b)). For



**Fig. 3.** (a) and (b): Minimal NIN-AND trees for 3 causes. (c) Op1. (d) Op2.

a larger number of causes, Theorem 1 suggests operations for automatic tree topology generation, which are illustrated in Fig. 3 (c) and (d).

**Theorem 1** *Every minimal NIN-AND tree for  $n \geq 3$  causes can be constructed by starting from an NIN-AND gate for 2 causes and applying a sequence of operations made of the following two:*

**Op1** *Add a root event as an input to a gate.*

**Op2** *Insert a new gate between a root event and an existing gate and add a new root event as the second input of the new gate.*

Proof by induction (sketch): For  $n = 3$ , there are exactly two minimal NIN-AND trees. The trees in Fig. 3 (a) and (b) can be constructed by applying Op1 and Op2 to the NIN-AND gate for 2 causes.

Assume that the theorem holds for  $n = k \geq 3$  and consider  $n = k + 1$ . Let  $T'$  be a minimal tree for  $n$  causes,  $x$  be a root event in  $T'$  and is connected to a gate  $g$ . Let  $T$  be a tree obtained by removing  $x$  from  $T'$ . Analyzing the following cases and applying the inductive assumption to  $T$ , it can be shown that, in each case,  $T'$  can be constructed by a sequence made of Op1 and Op2.

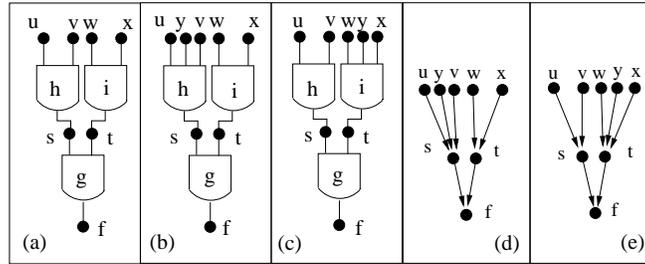
- Gate  $g$  has three or more input events.
- Gate  $g$  has two input events  $x$  and  $y$ .
  - Gate  $g$  is the leaf gate.
  - Gate  $g$  is a non-leaf gate.
    - \* Event  $y$  is a root event.
    - \* Event  $y$  is a non-root event.      $\square$

Theorem 1 suggests the following procedure to generate all minimal NIN-AND trees for  $n$  causes: Start with the NIN-AND gate for 2 causes. Apply Op1 and Op2 to the gate to generate the two NIN-AND trees for 3 causes. Repeat

the process to generate all minimal trees for 4 causes based on minimal trees for 3 causes, then those for 5 causes based on minimal trees for 4 causes, and so on, until those for  $n$  causes based on minimal trees for  $n - 1$  causes. At round  $k + 1$ , all minimal trees for  $k$  causes have been constructed from the round  $k$ . With each tree  $T$  for  $k$  causes, for each gate  $g$  in  $T$ , apply Op1 once to generate a tree for  $k + 1$  causes, and apply Op2 once to generate another tree if  $g$  has any root input event.

## 5 Removing Duplication Trees

The above procedure can generate all minimal NIN-AND trees for  $n$  causes. However, when executed, no matter what data structure is used, nodes in each tree is implicitly labeled. This causes generation of labeled trees corresponding to the same unlabeled NIN-AND tree. For instance, from the minimal tree in



**Fig. 4.** Trees in (b) and (c) are generated from (a). (d) Reduced tree from (b). (e) Reduced tree from (c).

Fig. 4 (a), when Op1 is applied to gate  $h$ , the labeled tree in (b) is generated. When the operation is applied to gate  $i$ , the tree in (c) is generated. In both (b) and (c), the new node is labeled  $y$ . However, in (b) it is connected to a gate adjacent to  $u, v$  and  $s$ . But in (c), it is connected to a gate adjacent to  $w, x$  and  $t$ . It is not obvious that trees in (b) and (c) correspond to the same unlabeled tree. To turn a list of generated trees into an enumeration, all labeled trees corresponding to the same unlabeled tree must be removed, except one.

Algorithm 2 below decides if a newly generated labeled NIN-AND tree  $T'$  corresponds to the unlabeled tree represented by an existing labeled tree  $T$ . It uses a more compact representation where each gate node is merged with the unique non-root event node connected to it. This converts a minimal NIN-AND tree to a *reduced tree*, defined below and illustrated in Fig. 4 (d) and (e).

**Def. 4** Let  $T$  be a minimal NIN-AND tree. A reduced tree relative to  $T$  is a directed graph obtained by (1) merging each gate node with the unique non-root event node that it is connected to, and (2) making all links explicitly directed in the same directions as in  $T$ .

Now the task is to recognize two reduced trees such as those in Fig. 4 (d) and (e) to be isomorphic. Two reduced trees are *isomorphic* if there is an one-to-one mapping of their nodes which (1) maps the leaf in one to the leaf in the other, and (2) preserves in-degree and out-degree. Algorithm 2 performs the task, where  $|T|$  denotes the number of nodes in  $T$ .

**Algorithm 2** *IsIsomorphicTree*( $T, T'$ )

*Input: Two reduced trees  $T$  and  $T'$ , with leaf nodes  $v$  and  $v'$ , and their parent sets  $\pi$  and  $\pi'$ , respectively.*

```

1  if  $|T'| \neq |T|$ , return false;
2  if  $|T'| = 1$ , return true;
3  if  $|\pi'| \neq |\pi|$ , return false;
4   $Sub' =$  the set of subtrees each with the leaf in  $\pi'$ ;
5   $Sub =$  the set of subtrees each with the leaf in  $\pi$ ;
6  for each subtree  $S' \in Sub'$ ,
7    for each unmarked subtree  $S \in Sub$ ,
8      if IsIsomorphicTree( $S, S'$ ) returns true,
9        mark  $S'$  and  $S$ ;
10     break inner for loop;
11  if  $S'$  is unmarked, return false;
12  return true;
```

Theorem 2 establishes that *IsIsomorphicTree* recognizes isomorphic reduced trees correctly.

**Theorem 2** *IsIsomorphicTree*( $T, T'$ ) returns true iff reduced trees  $T$  and  $T'$  are isomorphic.

Proof: We prove by strong induction on  $n = |T|$ . Denote  $n' = |T'|$ . For  $n = 1$ , the only case where  $T'$  and  $T$  are isomorphic is  $n' = 1$ . *IsIsomorphicTree* returns true in line 2. If  $T'$  and  $T$  are not isomorphic, we must have  $n' \neq 1$ . *IsIsomorphicTree* returns false in line 1.

Since  $T$  and  $T'$  are reduced NIN-AND trees,  $n > 1$  implies  $n \geq 3$  (similarly  $n' > 1$  implies  $n' \geq 3$ ). We therefore also consider the base case  $n = 3$ , where  $T$  has the leaf and two roots. If  $T$  and  $T'$  are not isomorphic, we must have  $n' \neq 3$ . *IsIsomorphicTree* returns false in line 1.  $T$  and  $T'$  are isomorphic whenever  $n' = 3$ . *IsIsomorphicTree* returns true in line 2.

Assume that the theorem holds for  $n = 1, 3, 4, \dots, k$  where  $k \geq 3$ . We consider the case  $n = k + 1$ . If  $T$  and  $T'$  are not isomorphic, either  $n' \neq n$ , or the degree of  $v'$  differs from that of  $v$ , or subtrees seated at parents of  $v'$  cannot be one-to-one mapped to subtrees seated at parents of  $v$  such that each pair is isomorphic. In the first two cases, *IsIsomorphicTree* returns false in lines 1 and 3. In the third case, for at least one subtree  $S'$ , no subtree  $S$  exists such that *IsIsomorphicTree* returns true by inductive assumption (the number of nodes in  $S$  is at most  $k - 1$ ). Hence, *IsIsomorphicTree* returns false in line 11.

On the other hand, if  $T$  and  $T'$  are identical, *IsIsomorphicTree* will enter the outer *for* loop. For each subtree  $S'$ , there exists a subtree  $S$  such that

IsIsomorphicTree returns true by inductive assumption (the number of nodes in  $S$  is at most  $k-1$ ). Hence, the outer *for* loop will complete and IsIsomorphicTree will return true in line 12.  $\square$

The following algorithm enumerates minimal NIN-AND trees of  $n \geq 3$  causes in terms of reduced trees.

**Algorithm 3** *EnumerateReducedTree( $n$ )*

```

1  initialize set  $ST_2$  with a single reduced tree for 2 causes;
2  for  $i = 3$  to  $n$ ,
3    initialize  $ST_i$  to empty set;
4    for each reduced tree  $T$  in  $ST_{i-1}$ ,
5      for each non-root node  $x$  of  $T$ ,
6        duplicate  $T$  as  $T'$ ;
7        add a root parent to  $x$  in  $T'$ ;
8        if for each  $S \in ST_i$ , IsIsomorphicTree( $S, T'$ ) returns false, add  $T'$  to  $ST_i$ ;
9        if  $x$  has a root parent  $p$ ;
10       duplicate  $T$  as  $T''$ ;
11       insert a new node  $z$  between  $x$  and  $p$  in  $T''$ ;
12       add a root parent to  $z$  in  $T''$ ;
13       if for each  $S \in ST_i$ , IsIsomorphicTree( $S, T''$ ) returns false,
           add  $T''$  to  $ST_i$ ;
14 return  $ST_n$ ;

```

The algorithm enumerates reduced trees of  $i$  causes based on the enumeration of  $i-1$  causes, collected in set  $ST_{i-1}$ . For each reduced tree  $T$  in  $ST_{i-1}$ , each non-root node  $x$  is processed, which corresponds to a gate of the original NIN-AND tree. For each  $x$ , Op1 and Op2 (if applicable) are applied. If the new tree is not isomorphic to one in set  $ST_i$ , it is added. Each reduced tree in  $ST_n$  can be easily converted to a minimal NIN-AND tree for  $n$  causes.

Due to Theorem 1 and one-to-one mapping between a reduced tree and the corresponding minimal NIN-AND tree, minimal NIN-AND trees corresponding to each  $ST_i$  are exhaustive. Due to Theorem 2, they are also mutually exclusive. Hence, we have the following theorem.

**Theorem 3**  $ST_n$  produced by *EnumerateReducedTree* enumerates minimal NIN-AND trees for  $n$  causes.

Using *EnumerateReducedTree*, we enumerated minimal NIN-AND trees for  $n$  causes with  $n$  between 2 and 10. The table below shows the execution result.

$n$	2	3	4	5	6	7	8	9	10
$ St_n $	1	2	5	12	33	90	261	766	2312

Our enumeration enables an NIN-AND tree for  $n$  causes to be elicited by displaying all alternative trees to human expert so that one can be selected. As it is often easier to select a target object from a list than to describe the object from vacuum, this technique is expected to improve the accuracy and efficiency in elicitation.

## 6 Pairwise Causal Interactions

Although selection from a list is usually less demanding mentally than description from vacuum, when the number of causes is more than six or seven, identifying the target NIN-AND tree accurately from the enumeration may still be challenging. For instance, there are 261 minimal NIN-AND trees for 8 causes. On the other hand, eliciting whether a pair of causes is reinforcing or undermining is much less demanding (binary selection). We therefore explore the possibility to uniquely identify a minimal NIN-AND tree based on elicitation of  $O(n^2)$  pairwise causal interactions.

To do so, we need to understand how a minimal NIN-AND tree determines a set of pairwise causal interactions. However, this is impossible because a minimal NIN-AND tree is unlabeled while a set of pairwise causal interactions must be specified over specific pairs of causes. On the other hand, any minimal NIN-AND tree with its root nodes labeled determines uniquely a set of pairwise causal interactions, as shown by the following proposition. We refer to such a tree as a minimal, *root-labeled* NIN-AND tree. Note that in a root-labeled tree, root nodes are labeled but non-root nodes are unlabeled. Note also the default assumption that every root node is a single-cause event.

**Proposition 4** *Let  $T$  be a minimal NIN-AND tree for a set  $X$  of causes and  $rl$  be a labeling of root nodes. Then  $T$  and  $rl$  define a function  $pci$  from pairs of distinct causes  $\{c_i, c_j\} \subset X$ , where  $i \neq j$ , to the set  $\{rif, udm\}$ , where  $rif$  stands for reinforcing and  $udm$  stands for undermining.*

For instance,  $pci(c_i, c_j) = udm$  means that causal interaction between  $c_i$  and  $c_j$  is undermining. Proof of Proposition 4 depends on Proposition 5 below, which in turn relies on the concept of the *closest common gate* (ccg):

**Def. 5** *Let  $x$  and  $y$  be two root nodes in a minimal NIN-AND tree  $T$ . Let  $path_x$  be the directed path from  $x$  to the leaf of  $T$  and  $path_y$  be that from  $y$ . Then, the first node in  $path_x$  common to a node in  $path_y$  is the ccg of  $x$  and  $y$ .*

In Fig. 4 (c), the ccg of  $x$  and  $y$  is  $i$ . Note that the first node common in  $path_x$  and  $path_y$  is always a gate. The following proposition shows how the interaction between a pair of causes is encoded in an NIN-AND tree.

**Proposition 5** *Let  $T$  be a minimal, root-labeled NIN-AND tree. Let  $x$  and  $y$  be a pair of root nodes,  $c_i$  and  $c_j$  be their corresponding causes, and  $g$  be their ccg. Then the causal interaction between  $c_i$  and  $c_j$  is of the type of  $g$ .*

Proof: Given root-labeled  $T$ , assume that all causes become inactive except  $c_i$  and  $c_j$ . Now except  $g$ , all other gates have no more than one active causal input event, and hence can be removed. The resultant minimal NIN-AND tree has a single gate  $g$  with root input events  $x$  and  $y$ .  $\square$

From Proposition 5, it is clear that the topology of  $T$  and a root labeling uniquely determine the type of causal interaction between each pair of causes (corresponding to a pair of roots in  $T$ ). Hence, Proposition 4 holds.

## 7 Are NIN-AND Trees PCI Differentiable?

Given that each minimal NIN-AND tree, plus a root labeling, uniquely determines a set of pairwise causal interactions, can a minimal NIN-AND tree be identified from a set of pairwise causal interactions? In other words, are pairwise causal interactions sufficient to differentiate minimal NIN-AND trees?

More specifically, let  $X = \{c_1, \dots, c_n\}$  be a set of  $n$  causes for an effect, and  $ST_n$  be the set of all minimal NIN-AND trees for  $n$  causes. Let  $pci$  be a function from pairs of distinct causes  $\{c_i, c_j\} \subset X$  ( $i \neq j$ ) to the set  $\{rif, udm\}$ , determined by  $T \in ST_n$  and a root labeling. The question then is whether it is possible to *uniquely* identify  $T$  given  $pci$  only. For instance, the NIN-AND tree in Fig. 2 (a) has  $pci(c_1, c_2) = pci(c_1, c_3) = pci(c_2, c_3) = udm$ ,  $pci(c_1, c_4) = pci(c_2, c_4) = pci(c_3, c_4) = rif$ . Can the tree model be uniquely identified from the function? We refer to this as the *identification* question.

Note that since  $T$  and a root labeling together define a causal interaction function  $pci$ , to answer the above question in general, the search space is not  $ST_n$  but the space of all root-labeled minimal NIN-AND trees, whose complexity is  $O(n! |ST_n|)$ .

We answer the identification question by exhaustively testing whether there exists a pair of NIN-AND trees  $T$  and  $T'$  and there exists a root labeling for each of them, such that the two root-labeled trees satisfy the same set of pairwise causal interactions corresponding to some  $pci$  function. Although this method does not scale for very large  $n$ , we point out that a NIN-AND tree model is used to acquire a single CPT in a BN and hence very large  $n$  is not expected given the conditional independence expressed by the BN. To make the testing computation effective, we developed the following test conditions:

Given  $n$  causes, there are  $n(n-1)/2$  pairs. A  $pci$  function maps each pair to one of  $rif$  and  $udm$ , thus defining two sets of pairs which we refer to as  $Rif$  and  $Udm$ . That is, a pair  $\{c_i, c_j\} \in Udm$ , iff  $pci(c_i, c_j) = udm$ . Once one of  $Rif$  and  $Udm$  is defined, the other is uniquely determined.

Proposition 6 below shows that if two minimal NIN-AND trees, under arbitrary root-labeling, produce two  $Rif$  sets of different cardinalities, then the two trees are differentiable from pairwise casual interactions.

**Proposition 6** *Let  $T$  and  $T'$  be two minimal NIN-AND trees of  $n$  causes,  $rl$  and  $rl'$  be some root labeling of  $T$  and  $T'$ , and  $pci$  and  $pci'$  be the corresponding pairwise causal interaction functions, respectively. Let  $Rif$  and  $Rif'$  be the sets of reinforcing pairs, defined by  $pci$  and  $pci'$ , respectively.*

*If  $|Rif| \neq |Rif'|$ , there exist no root labeling for  $T$  and  $T'$ , such that the two root-labeled trees satisfy the same pairwise causal interaction function.*

Proof: When  $|Rif| \neq |Rif'|$ , the number of reinforcing cause pairs defined by  $T$  and  $rl$  differs from that defined by  $T'$  and  $rl'$ . Since the number of reinforcing cause pairs is independent of root labeling, no matter what alternative root labeling are used, the inequality remains. This implies that  $Rif \neq Rif'$  no matter what root labeling is used. Hence, the proposition holds.  $\square$

Proposition 6 suggests an inexpensive test (to be referred as Test 1) that rules out the tree pair from possibly contributing to the negative answer of identification question.

If two minimal NIN-AND trees, under some root-labeling, produce *Rif* sets of identical cardinalities, it is uncertain how the tree pair contributes to the answer of identification question. The following proposition suggests a further test.

**Proposition 7** *Let  $T, T', rl, rl', pci$  and  $pci'$  be defined as in Proposition 6. For each cause  $c$ , let  $k(c)$  be the number of other causes that are pairwise reinforcing with  $c$  according to  $pci$ , and  $\kappa$  be the sorted list of  $k(c)$ 's. Let  $\kappa'$  be the corresponding sorted list defined by  $pci'$ .*

*If  $\kappa \neq \kappa'$ , there exist no root labeling for  $T$  and  $T'$ , such that the two root-labeled trees satisfy the same pairwise causal interaction function.*

For the NIN-AND tree in Fig. 2 (a),  $\kappa = (1, 1, 1, 3)$  because  $k(c_1) = k(c_2) = k(c_3) = 1$  and  $k(c_4) = 3$ .

Proof: If  $\kappa \neq \kappa'$ , then there exists  $k$  ( $0 \leq k < n(n-1)/2$ ) such that  $m$  elements of  $\kappa$  have value  $k$ ,  $m'$  elements of  $\kappa'$  have value  $k$ , and  $m \neq m'$ . That is, according to  $T$  under root-labeling  $rl$ , each of  $m$  causes reinforces with another  $k$  causes. But according to  $T'$  under root-labeling  $rl'$ , each of  $m'$  causes reinforces with another  $k$  causes.

The number of other causes which a given cause reinforces with is independent of root labeling. Hence, no matter what alternative root labeling are used, according to  $T$  the number of causes reinforcing with  $k$  other causes remains  $m$ , and according to  $T'$  the number remains  $m'$ . This implies that  $\kappa \neq \kappa'$  no matter what root labeling is used. Hence, the proposition holds.  $\square$

Proposition 7 suggests another inexpensive test (to be referred as Test 2) that rules out a tree pair from possibly contributing to the negative answer of identification question. Algorithm 4 utilizes Test 1 and Test 2 in answering identification question.

Lines 1 through 4 enumerate minimal NIN-AND trees and compute the *pci* function for each tree under some root labeling. Lines 6 and 7 perform Test 1, and lines 8 and 9 perform Test 2. Lines 10 through 12 (Test 3) tries each of the  $n! - 1$  alternative root labeling on  $T'$ . If one produces *Rif'* identical to *Rif* for  $T$ , then  $T$  and  $T'$  cannot be differentiated by pairwise causal interactions. The algorithm will return false.

**Algorithm 4** *IsPciIdentifiable( $n$ )*

- 1 *run EnumerateReducedTree( $n$ ) to produce  $ST_n$ ;*
- 2 *convert reduced trees in  $ST_n$  to minimal NIN-AND trees;*
- 3 *for each minimal NIN-AND tree  $T$ ,*
- 4 *compute its *pci* function under some root labeling;*
- 5 *for each pair of trees  $T$  and  $T'$  with *pci* and *pci'*,*
- 6 *compute *Rif* and *Rif'* from *pci* and *pci'*;*

```

7   if  $|Rif| \neq |Rif'|$ , go to line 5 for next pair;
8   compute  $\kappa$  and  $\kappa'$  from  $pci$  and  $pci'$ ;
9   if  $\kappa \neq \kappa'$ , go to line 5 for next pair;
10  for each alternative root labeling  $rl'$  of  $T'$ ,
11    recompute  $Rif'$ ;
12    if  $Rif = Rif'$ , return false;
13  return true;

```

If for every pair of  $T$  and  $T'$ , either Test 1 fails ( $|Rif| \neq |Rif'|$ ), or Test 2 fails ( $\kappa \neq \kappa'$ ), or Test 3 fails ( $Rif \neq Rif'$  for all root labeling), the algorithm returns true. This means that every minimal NIN-AND tree for  $n$  causes can be identified based solely on a set of pairwise causal interactions. This leads to the following theorem, whose proof is straightforward given the above analysis.

**Theorem 4** *Let  $T$  be a minimal NIN-AND tree for  $n$  causes and  $Rif$  is the set of pairwise causal interactions determined by  $T$  under some root labeling. Then given  $Rif$  only,  $T$  can be identified from all minimal NIN-AND trees for  $n$  causes iff algorithm  $IsPciIdentifiable(n)$  returns true.*

For example, given  $Rif = \{\{c_1, c_4\}, \{c_2, c_4\}, \{c_3, c_4\}\}$ , the minimal NIN-AND tree in Fig. 2 (a) is uniquely identified. Justified by Theorem 4, we implemented  $IsPciIdentifiable(n)$ . Executions for  $n = 3, \dots, 10$  all returned true. Tests 1 and 2 suggested by Propositions 6 and 7 trim computation significantly. For  $n = 10$ , there are 2312 minimal NIN-AND trees, and hence 2671516 pairs. Only 122588 pairs ( $< 5\%$ ) passed the least expensive Test 1, where Test 2 is needed. Out of these pairs, only 467 pairs ( $< 0.4\%$ ) passed Test 2, where the most expensive Test 3 has to be run. This amounts to the processing of 1694649600 root-labeled trees in Test 3, which is about 20% of the total 8389785600 root-labeled trees.

## 8 Remarks

Assessment of CPTs is often a bottleneck in practical applications of BNs when frequency data are not available and elicitation from expert is necessary. This work follows the effort by many, e.g., [7, 4, 3, 5, 9], to make this step in probabilistic reasoning more efficient. The main contributions are the following:

From associativity of NIN-AND gates, we characterized minimal NIN-AND tree topology space and partitioned it into the subspace with direct leaf NIN-AND gates and the subspace with dual leaf gates. This partition allows subsequent investigation to be focused on one subspace while the results are applicable to the other. We developed a method to enumerate NIN-AND trees based on local insertion. This result provides an alternative method for NIN-AND tree elicitation and allows expert to select from enumeration. We demonstrated that NIN-AND trees for up to 10 causes (about the necessary upper bound for CPTs in BNs) can be differentiated based on pairwise causal interactions. This result provides an even more powerful technique for eliciting NIN-AND trees by eliciting from expert only pairwise causal interactions.

In developing the two elicitation techniques, enumeration of NIN-AND trees by number of causes (Section 4) is needed. As information flows from single-cause events to the multi-cause event in an NIN-AND tree, following the convention in causal graphical models, we directed reduced NIN-AND trees with single-cause events as roots and with a single leaf. We therefore needed to enumerate unlabeled trees of a single leaf by number of roots. Many methods of tree enumeration in the mathematics literature, e.g., [1, 8, 6], do not address this problem. In [2], under the context of phylogenetic trees, counting of *rooted multifurcating tree shapes by tips* is presented. Reversing directions of links, these tree shapes are equivalent to what we enumerate. However, Felsenstein’s counting is based on an ordered partition of tips, and our method is based on local insertion. It focuses on *counting* without generation and ours emphasizes generation. In [10], we extend [2] into a method as an alternative to the method presented here.

## Acknowledgements

We acknowledge the financial support through the Discovery Grant, NSERC, Canada. We thank anonymous reviewers for their helpful comments.

## References

1. A. Cayley. A theorem on trees. *Quarterly J. Mathematics*, pages 376–378, 1889.
2. J. Felsenstein. *Inferring Phylogenies*. Sinauer Associates, Sunderland, Mass., 2004.
3. S.F. Galan and F.J. Diez. Modeling dynamic causal interaction with Bayesian networks: temporal noisy gates. In *Proc. 2nd Inter. Workshop on Causal Networks*, pages 1–5, 2000.
4. D. Heckerman and J.S. Breese. Causal independence for probabilistic assessment and inference using Bayesian networks. *IEEE Trans. on System, Man and Cybernetics*, 26(6):826–831, 1996.
5. J.F. Lemmer and D.E. Gossink. Recursive noisy OR - a rule for estimating complex probabilistic interactions. *IEEE Trans. on System, Man and Cybernetics, Part B*, 34(6):2252–2261, 2004.
6. J.W Moon. *Counting Labeled Trees*. William Clowes and Sons, London, 1970.
7. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
8. J. Riordan. The enumeration of trees by height and diameter. *IBM J.*, pages 473–478, Nov. 1960.
9. Y. Xiang and N. Jia. Modeling causal reinforcement and undermining for efficient cpt elicitation. *IEEE Trans. Knowledge and Data Engineering*, 19(12):1708–1718, 2007.
10. Y. Xiang, J. Zhu, and Y. Li. Enumerating unlabeled and root labeled trees for causal model acquisition. Springer-Verlag. To appear in Proc. Canadian AI 2009.