

A Simple Method for Testing Independencies in Bayesian Networks

Cory J. Butz¹(✉), André E. dos Santos¹, Jhonatan S. Oliveira¹,
and Christophe Gonzales²

¹ Department of Computer Science, University of Regina, Regina, Canada
{butz,dossantos,oliveira}@cs.uregina.ca

² LIP6 - département DESIR, Université Pierre Et Marie Curie, Paris, France
christophe.gonzales@lip6.fr

Abstract. Testing independencies is a fundamental task in reasoning with *Bayesian networks* (BNs). In practice, *d-separation* is often utilized for this task, since it has linear-time complexity. However, many have had difficulties in understanding *d-separation* in BNs. An equivalent method that is easier to understand, called *m-separation*, transforms the problem from directed separation in BNs into classical separation in undirected graphs. Two main steps of this transformation are pruning the BN and adding undirected edges.

In this paper, we propose *u-separation* as an even simpler method for testing independencies in a BN. Our approach also converts the problem into classical separation in an undirected graph. However, our method is based upon the novel concepts of inaugural variables and rationalization. Thereby, the primary advantage of *u-separation* over *m-separation* is that *m-separation* can prune unnecessarily and add superfluous edges. Hence, *u-separation* is a simpler method in this respect.

Keywords: Bayesian networks · Testing independencies · *d-separation* · *m-separation*

1 Introduction

Pearl [9] states that perhaps the founding of *Bayesian networks* (BNs) [3,4,7] made its greatest impact through the notion of *d-separation*. *d-Separation* [8] is a graphical method for deciding which conditional independence relations are implied by the *directed acyclic graph* (DAG) of a BN. To test whether two sets X and Z of variables are conditionally independent given a third set Y of variables, denoted $I(X, Y, Z)$, *d-separation* checks whether every path from X to Z is “blocked” by Y in the DAG. *d-Separation* considers every variable on each of these paths. Each variable is classified into one of three categories. The same variable can assume different classifications depending on which path is being considered. Depending on the direction of the edges, sometimes blocking works intuitively. On the contrary, sometimes a path is not “blocked” by Y even though it necessarily traverses Y . Unfortunately, the above has led to many

having difficulties in understanding d-separation [10] even though there exists a linear-time implementation of d-separation [2].

m-Separation [5, 13], an equivalent method for testing independencies in a BN, seeks to avoid the confusion associated with the directionality of edges in a DAG by turning the problem into classical separation in an undirected graph. The undirected graph is constructed using a two-step process. First, a sub-DAG is built by pruning the DAG of the BN. Second, the constructed sub-DAG is moralized. The *moralization* [6] of a DAG involves adding an undirected edge between every pair of variables with a common child and then dropping directionality. The independence $I(X, Y, Z)$ holds in the BN if and only if X and Z are separated by Y in the undirected graph.

The main contribution of this paper is an even simpler method for testing independencies in BNs, called *undirected separation* (*u-separation*). Like *m-separation*, we seek to apply classical separation in an undirected graph built from the given BN. We prove that only *inaugural* [1] variables need be pruned from the BN. Next, we suggest the notion of *rationalization* as a refinement of moralization. Unlike moralization, rationalization only adds an undirected edge between parents of a common child when the common child is in Y of $I(X, Y, Z)$. We establish that *u-separation* is equivalent to *m-separation*. Two advantages of *u-separation* are that it can prune fewer variables than *m-separation* and add fewer undirected edges than *m-separation*. Given the prominence of d-separation [9], the novel method of *u-separation* provides a clearer understanding of BNs.

This paper is organized as follows. In Sect. 2, *m-separation* is reviewed. Section 3 presents *u-separation*. The equivalence between *m-separation* and *u-separation* is established in Sect. 4. In Sect. 5, advantages of *u-separation* are provided. Conclusions are given in Sect. 6.

2 Background

Let $U = \{v_1, v_2, \dots, v_n\}$ be a finite set of variables. Let \mathcal{B} be a *directed acyclic graph* (DAG) on U . A *directed path* from v_1 to v_k is a sequence v_1, v_2, \dots, v_k with directed edges (v_i, v_{i+1}) in \mathcal{B} , $i = 1, 2, \dots, k - 1$. For each $v_i \in U$, the *ancestors* of v_i , denoted $An(v_i)$, are those variables having a directed path to v_i , while the *descendants* of v_i , denoted $De(v_i)$, are those variables to which v_i has a directed path. For a set $X \subseteq U$, we define $An(X)$ and $De(X)$ in the obvious way. The *children* $Ch(v_i)$ and *parents* $Pa(v_i)$ of v_i are those v_j such that $(v_i, v_j) \in \mathcal{B}$ and $(v_j, v_i) \in \mathcal{B}$, respectively. An *undirected path* in a DAG is a path ignoring directions. A *path* in an undirected graph is defined similarly. The *moralization* [6] of a DAG is the undirected graph formed by adding an undirected edge between each pair of parents of a common child and then dropping directionality. A singleton set $\{v\}$ may be written as v , $\{v_1, v_2, \dots, v_n\}$ as $v_1 v_2 \dots v_n$, and $X \cup Y$ as XY .

A *Bayesian network* (BN) [3, 4, 7] is a DAG \mathcal{B} on U together with *conditional probability tables* (CPTs) $P(v_1|Pa(v_1))$, $P(v_2|Pa(v_2))$, \dots , $P(v_n|Pa(v_n))$. For example, Fig. 1 shows a BN, where CPTs $P(a)$, $P(b)$, \dots , $P(j|i)$ are not provided.

We call \mathcal{B} a BN, if no confusion arises. The product of the CPTs for \mathcal{B} on U is a *joint probability distribution* $P(U)$ [7]. The *conditional independence* [12] of X and Z given Y holding in $P(U)$ is denoted $I_P(X, Y, Z)$. It is known that if $I(X, Y, Z)$ holds in \mathcal{B} , then $I_P(X, Y, Z)$ holds in $P(U)$.

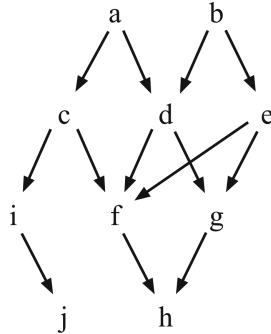


Fig. 1. A BN \mathcal{B} modified from [3].

m-Separation [5, 13] is an equivalent method to *d-separation* [7] for testing independencies in BNs. Let $X, Y,$ and Z be pairwise disjoint sets of variables in a BN \mathcal{B} . *m-Separation* tests $I(X, Y, Z)$ with four steps: (i) prune \mathcal{B} onto $XYZ \cup An(XYZ)$; (ii) construct the moralization of the sub-DAG from (i); (iii) delete Y and its incident edges from the undirected graph built in (ii); and (iv) if there exists a path from (any variable in) X to (any variable in) Z in (iii), then $I(X, Y, Z)$ does not hold; otherwise, $I(X, Y, Z)$ holds.

Example 1. Consider testing $I(a, de, g)$ using *m-separation* in the BN \mathcal{B} of Fig. 1. In step (i), the sub-DAG constructed on $\{a, d, e, g\} \cup An(\{a, d, e, g\}) = \{a, b, d, e, g\}$ is illustrated in Fig. 2(i). In step (ii), the moralization of the sub-DAG is depicted in Fig. 2(ii), where undirected edges (a, b) and (d, e) were added and then directionality was dropped. In step (iii), variables d and e and their

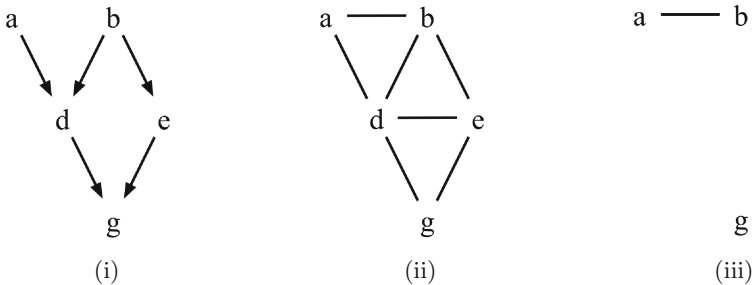


Fig. 2. When testing $I(a, de, g)$ in the BN \mathcal{B} in Fig. 1, *m-separation* builds the sub-DAG in (i), determines the moralization in (ii), and deletes $Y = \{d, e\}$ and the incident edges, giving (iii).

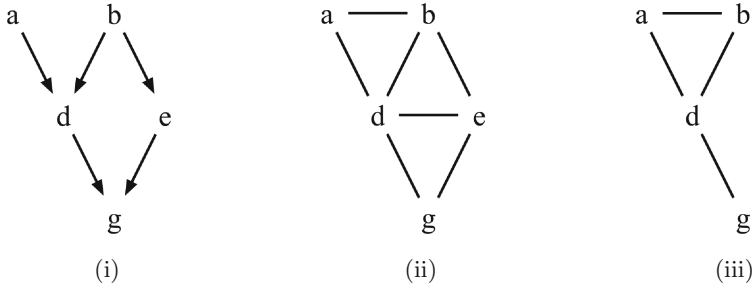


Fig. 3. When testing $I(a, e, g)$ in the BN \mathcal{B} in Fig. 1, m-separation builds the sub-DAG in (i), determines the moralization in (ii), and deletes $Y = \{e\}$ and the incident edges, yielding (iii).

incident edges (a, d) , (b, d) , (b, e) , (d, e) , (d, g) , and (e, g) are deleted, yielding the undirected graph in Fig. 2(iii). Since there does not exist a path from a to g in Fig. 2(iii), $I(a, de, g)$ holds in \mathcal{B} by m-separation.

Example 2. Consider testing $I(a, e, g)$ using m-separation in the BN \mathcal{B} of Fig. 1. In step (i), the sub-DAG constructed is illustrated in Fig. 3(i). In step (ii), the moralization of the sub-DAG is depicted in Fig. 3(ii), where undirected edges (a, b) and (d, e) were added and then directionality was dropped. In step (iii), variable e and the incident edges (b, e) , (d, e) , and (e, g) are deleted, giving Fig. 3(iii). Since there exists a path from a to g in Fig. 3(iii), $I(a, e, g)$ does not hold in \mathcal{B} by m-separation.

3 u-Separation

Our simplification of m-separation, called *undirected separation (u-separation)*, requires the notion of inaugural variables and a refinement of moralization.

A variable v_k is called a *v-structure* [11] in a BN \mathcal{B} , if \mathcal{B} contains directed edges (v_i, v_k) and (v_j, v_k) , but not a directed edge between variables v_i and v_j in \mathcal{B} . For example, variable f is a v-structure in BN \mathcal{B} of Fig. 1, since \mathcal{B} contains directed edges (d, f) and (e, f) , for instance, and does not contain either directed edges (d, e) or (e, d) .

Given an independence $I(X, Y, Z)$ to be tested in a BN \mathcal{B} , a variable v is *inaugural* [1], if at least one of the following two conditions is satisfied: (i) v is a v-structure and

$$(\{v\} \cup De(v)) \cap XYZ = \emptyset; \tag{1}$$

or (ii) v is a descendant of a variable satisfying (i). With respect to a given independence $I(X, Y, Z)$ being tested against a BN \mathcal{B} , we denote by V the set of all inaugural variables.

Example 3. Consider testing $I(a, de, g)$ in the BN \mathcal{B} of Fig. 1. Variable f is inaugural, since it is a v-structure and, by (1),

$$(\{f\} \cup \{h\}) \cap \{a, d, e, g\} = \emptyset.$$

Consequently, by condition (ii), h is also inaugural, since h is a descendant of f . On the contrary, variable d is a v-structure, but is not inaugural, since

$$(\{d\} \cup \{f, g, h\}) \cap \{a, d, e, g\} \neq \emptyset.$$

Therefore, the set of all inaugural variables in \mathcal{B} for $I(a, de, g)$ is $V = \{f, h\}$.

Definition 1. The *rationalization* of a DAG with respect to an independence $I(X, Y, Z)$ is the undirected graph constructed by adding an undirected edge between parents of a common child v , if $v \in Y$, and then dropping directionality.

Example 4. Given the DAG in Fig. 4(i) and the independence $I(a, de, g)$, the rationalization is in Fig. 4(ii). Here, an undirected edge (a, b) was added between variables a and b , since common child $d \in Y = \{d, e\}$. Directionality was then dropped.



Fig. 4. When testing $I(a, de, g)$ in the DAG in (i), rationalization gives (ii).

Note the key difference between moralization and rationalization. Moralization of Fig. 2(i) adds undirected edges (a, b) and (d, e) as shown in Fig. 2(ii). In contrast, rationalization of Fig. 4(i) only adds undirected edge (a, b) in Fig. 4(ii).

We now formally introduce u-separation.

u-Separation tests independencies in BNs. Let $X, Y,$ and Z be pairwise disjoint sets of variables in a BN \mathcal{B} . Then, u-separation tests $I(X, Y, Z)$ with four steps: (i) prune all inaugural variables from \mathcal{B} ; (ii) construct the rationalization of the sub-DAG in (i); (iii) delete Y and its incident edges from the undirected graph built in (ii); and (iv) if there exists a path from (any variable in) X to (any variable in) Z in (iii), then $I(X, Y, Z)$ does not hold; otherwise, $I(X, Y, Z)$ holds.

Example 5. Let us test $I(a, de, g)$ in the BN \mathcal{B} of Fig. 1 using u-separation. By Example 3, f and h are the only inaugural variables. Thus, step (i) of u-separation prunes inaugural variables f and h from \mathcal{B} , yielding the sub-DAG

in Fig. 5(i). In step (ii), the rationalization of the constructed sub-DAG gives the undirected graph in Fig. 5(ii). Variables d and e and their incident edges are deleted in step (iii), yielding Fig. 5(iii). In step (iv), there does not exist a path from a to g . Thus, $I(a, de, g)$ holds in \mathcal{B} by u-separation.

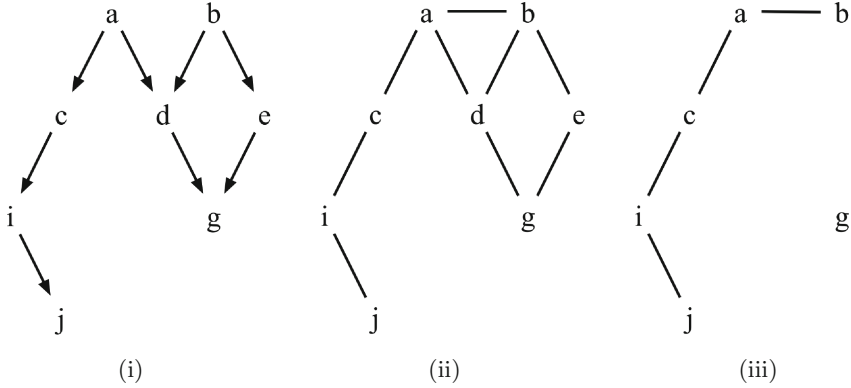


Fig. 5. When testing $I(a, de, g)$ in the BN \mathcal{B} in Fig. 1, u-separation builds the sub-DAG in (i) by pruning all inaugural variables, determines the rationalization in (ii), and deletes $Y = \{d, e\}$ and the incident edges, yielding (iii).

Example 6. Let us test $I(a, e, g)$ in the BN \mathcal{B} of Fig. 1 using u-separation. Step (i) of u-separation prunes inaugural variables f and h from \mathcal{B} , yielding the sub-DAG in Fig. 6(i). In step (ii), the rationalization of the constructed sub-DAG yields the undirected graph in Fig. 6(ii). Variable e and its incident edges are deleted in step (iii), yielding Fig. 6(iii). Since there exists a path from a to g in Fig. 6(iii), $I(a, e, g)$ does not hold in \mathcal{B} by u-separation.

4 Equivalence of u-Separation and m-Separation

We introduce pertinent notation to establish the equivalence of u-separation and m-separation. Given independence $I(X, Y, Z)$ to be tested in a BN \mathcal{B} on U , we define the following set:

$$W = U - (XYZ \cup An(XYZ)). \tag{2}$$

In other words, W is the set of all variables pruned from \mathcal{B} when m-separation builds the sub-DAG in step (i).

Lemma 1. *Given $I(X, Y, Z)$ to be tested in a BN \mathcal{B} . Let V be the set of all inaugural variables. Let W be the set of all variables pruned by m-separation in step (i). Then, $V \subseteq W$.*

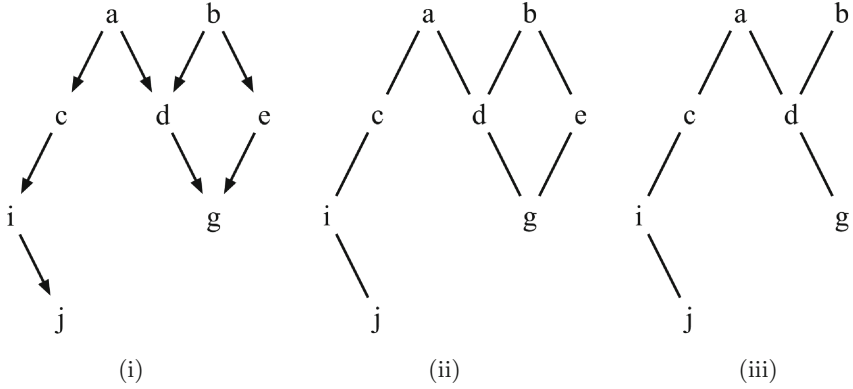


Fig. 6. When testing $I(a, e, g)$ in the BN \mathcal{B} in Fig. 1, u-separation builds the sub-DAG in (i) by pruning all inaugural variables, determines the rationalization in (ii), and deletes $Y = \{e\}$ and the incident edges, yielding (iii).

Proof. Let $v \in V$. By (1), $(v \cup De(v)) \cap XYZ = \emptyset$. Thus, $v \cap XYZ = \emptyset$ and $De(v) \cap XYZ = \emptyset$. Now, $De(v) \cap XYZ = \emptyset$ implies that $v \notin An(XYZ)$. Thus, as $v \notin XYZ$ and $v \notin An(XYZ)$, $v \in W$.

Lemma 1 implies that every variable pruned by u-separation is also pruned by m-separation.

Example 7. Given $I(a, de, g)$ to be tested in the BN \mathcal{B} of Fig. 1, step (i) of u-separation constructs the sub-DAG in Fig. 5(i) by pruning $V = \{f, h\}$. On the other hand, step (i) of m-separation builds the sub-DAG in Fig. 2(i) by pruning $W = \{c, f, i, j, h\}$. By Lemma 1, $V \subseteq W$.

Now, we consider the set of variables pruned by m-separation but not by u-separation. Given $I(X, Y, Z)$ to be tested in a BN \mathcal{B} . Let V be the set of all inaugural variables. Let W be the set of all variables pruned by m-separation in step (i). Then the set S of variables pruned by m-separation but not by u-separation is defined as:

$$S = W - V. \tag{3}$$

Lemma 2. *Given $I(X, Y, Z)$ to be tested in a BN \mathcal{B} . Then $S \cap (XYZ \cup An(XYZ)) = \emptyset$, where S is defined in (3).*

Proof. By (2), $W \cap (XYZ \cup An(XYZ)) = \emptyset$. Now, by (3), $S \subseteq W$. Hence, the claim follows.

Recall that S denotes the set of variables pruned by m-separation but not by u-separation. Lemma 2 means that none of these variables appear in $I(X, Y, Z)$ nor are ancestors of variables therein.

Example 8. Consider W and V from Example 7 when testing $I(a, de, g)$ in the BN \mathcal{B} of Fig. 1. Then, $V = \{f, h\}$, $W = \{c, f, i, j, h\}$, and $S = \{c, i, j\}$. Moreover, $\{a, d, e, g\} \cup An(\{a, d, e, g\}) = \{a, b, d, e, g\}$. As promised by Lemma 2, $\{c, i, j\} \cap \{a, b, d, e, g\} = \emptyset$.

Lemma 3. *Given $I(X, Y, Z)$ to be tested in a BN \mathcal{B} , let S be defined as in (3). Then no variable in S is a v-structure in \mathcal{B} .*

Proof. By contradiction, suppose $v \in S$ is a v-structure in \mathcal{B} . By (3), $v \notin V$. Then, by (1), $(v \cup De(v)) \cap XYZ \neq \emptyset$. Thus, either

$$v \cap XYZ \neq \emptyset \tag{4}$$

or

$$De(v) \cap XYZ \neq \emptyset, \tag{5}$$

or both, hold. Suppose (4) holds. Then $v \in XYZ$. By Lemma 2, $S \cap (XYZ \cup An(XYZ)) = \emptyset$. A contradiction, since by assumption $v \in S$. Now suppose (5) holds. Then there exists a $v' \in De(v)$ such that $v' \in XYZ$. Since $v \in An(v')$, we know $v \in An(XYZ)$. A contradiction to Lemma 2, since by assumption $v \in S$.

Lemma 3 implies that there are no v-structures among the variables pruned by m-separation but not by u-separation.

Example 9. Consider $S = \{c, i, j\}$ from Example 8 when testing $I(a, de, g)$ in the BN \mathcal{B} of Fig. 1. No variable in S is a v-structure in \mathcal{B} .

We now show the equivalence of u-separation and m-separation.

Theorem 1. *Testing independence $I(X, Y, Z)$ in a BN \mathcal{B} with m-separation is equivalent to testing independence $I(X, Y, Z)$ in \mathcal{B} with u-separation.*

Proof. We establish the equivalence between m-separation and u-separation by showing the equivalence of their corresponding steps.

Consider the sub-DAGs built in step (i) of m-separation and u-separation. All paths are the same between the two sub-DAGs, except for those involving variables in S . We now show that these paths do not affect the result of testing $I(X, Y, Z)$. By contradiction, suppose there is an undirected path from X to Z involving S . Then there exists two undirected edges (v_1, s_1) and (v_2, s_2) on this path, where $s_1, s_2 \in S$ and $v_1, v_2 \notin S$. Note that s_1 and s_2 may be the same variable. Now (v_1, s_1) must be a directed edge in \mathcal{B} . Otherwise, s_1 would be parent of v_1 , and thus would not be pruned by step (i) in m-separation. Similarly, (v_2, s_2) also must be a directed edge in \mathcal{B} . This immediately implies that there exists a variable s_3 on the path such that $s_3 \in S$ and s_3 is a v-structure in \mathcal{B} . A contradiction to Lemma 3. Therefore, there does not exist an undirected path from X to Z going through S .

Now consider the undirected edges added to the sub-DAGs in step (ii) of m-separation and u-separation. By Lemma 3, S does not contain v-structures in \mathcal{B} .

Therefore, the set of v-structures considered is the same for both m-separation and u-separation. For any v-structure v with $v \in Y$ of $I(X, Y, Z)$, both rationalization and moralization will add an edge connecting its parents. The only difference is the case when v-structure v is not in Y . This means variable v is not deleted in step (iii). Let (v_1, v) and (v_2, v) be directed edges in \mathcal{B} , namely, v_1 and v_2 are parents of v in \mathcal{B} , with no directed edge between v_1 and v_2 . There are two cases to consider. Suppose that neither v_1 nor v_2 are in Y . This means that v_1 and v_2 both are not deleted in step (iii). Hence, adding edge (v_1, v_2) is redundant, since there already is an undirected path $(v_1, v), (v, v_2)$ from v_1 to v_2 . Now suppose that at least one of v_1 and v_2 are in Y . In this case, either v_1 or v_2 , or both, are deleted in step (iii) along with their incident edges. Thus, adding edge (v_1, v_2) is wasteful in step (ii), since it will be deleted in step (iii). Therefore, in either case, adding (v_1, v_2) is immaterial.

Steps (iii) and (iv) are the same in both methods.

Theorem 1 establishes that testing independence with u-separation is equivalent to testing independence with m-separation.

Example 10. Consider the BN \mathcal{B} in Fig. 1. $I(a, de, g)$ holds by m-separation in Example 1 and holds by u-separation in Example 5. Similarly, $I(a, e, g)$ does not hold by m-separation in Example 2 and does not hold by u-separation in Example 6.

5 Advantages

Salient features of u-separation compared to m-separation are described.

In its attempt to transform directed separation in DAGs into classical separation in undirected graphs, m-separation can be heavy-handed in two aspects.

First, in step (i), m-separation can prune variables from the BN unnecessarily. Variables not affecting the outcome of the separation test do not need be pruned. Doing so is wasteful.

Example 11. Consider testing $I(a, de, g)$ in the BN \mathcal{B} of Fig. 1 using m-separation. Here, variables $\{c, f, h, i, j\}$ are pruned, yielding the sub-DAG in Fig. 2(i). Closer inspection reveals that variables $c, i,$ and j would not affect the separation test between variables a and g . Thus, it was wasteful of m-separation to prune these three variables.

u-Separation corrects this shortcoming by pruning only inaugural variables from the BN. In Example 11, u-separation only prunes variables $\{f, h\}$ instead of $\{c, f, h, i, j\}$.

The second undesirable characteristic of m-separation is that moralization can be excessive. More specifically, moralization may add edges to the undirected graph that have no bearing on the outcome of the separation test.

Example 12. Recall how m-separation tests $I(a, de, g)$ in Example 1. Step (ii) of m-separation includes adding undirected edge (d, e) , while step (iii) involves deleting edge (d, e) .

Example 13. Recall how m-separation tests $I(a, e, g)$ in Example 2. Step (ii) of m-separation involves adding undirected edge (a, b) . However, edge (a, b) is immaterial in Fig. 3(ii), since edges (a, d) and (b, d) are also present in step (iv).

Closer inspection of Example 2 highlights the excessiveness of m-separation using moralization to test independencies. Moralization adds two edges, (a, b) and (d, e) in Fig. 3(ii). Edge (d, e) is subsequently deleted by m-separation, while edge (a, b) is superfluous.

u-Separation remedies this failing by introducing the notion of rationalization. Rationalization is a refinement of moralization, since the edges added by rationalization are a subset of those added by moralization. When testing $I(X, Y, Z)$, whereas m-separation adds an undirected edge between variables with a common child, u-separation does so only when the child is in Y . This means that the v-structure variable will be subsequently deleted and the added edge between the parents is required to encode the dependence between them. In Example 5, for instance, u-separation only adds edge (a, b) rather than edges (a, b) and (d, e) .

It is perhaps worth mentioning here that moralization is ideally suited when applied for building a *join tree* [6] of a given DAG. For this purpose, moralization ensures that a clique containing all variables in each CPT of the BN is formed in the undirected graph. On the contrary, moralization can be overkill when applied for testing independencies in BN.

6 Conclusion

We have suggested *u-separation* as a new method for testing independencies in BNs. Our emphasis here, like that of m-separation, is on simplicity rather than speed. We have demonstrated how u-separation is simpler than m-separation. By utilizing the notion of *inaugural* variables, Lemma 1 ensures that u-separation will never prune more variables than m-separation. This means that m-separation can prune variables unnecessarily to apply separation in undirected graphs, as Example 7 shows.

Moreover, our analysis shows how m-separation's use of moralization is wasteful. An edge added in step (ii) can be deleted in step (iii), as illustrated in Example 12. Furthermore, Example 13 highlights how m-separation can add an edge that is extraneous. To overcome this excessiveness, u-separation suggests the use of *rationalization* rather than moralization. Rationalization is better suited for testing independencies in BNs, whereas moralization is perfectly suited for building a join tree in BN inference.

References

1. Butz, C.J., dos Santos, A.E., Oliveira, J.S., Gonzales, C.: Testing independencies in Bayesian networks with i-Separation. In: Proceedings of the Twenty-Ninth International FLAIRS Conference (2016)

2. Geiger, D., Verma, T.S., Pearl, J.: d-separation: from theorems to algorithms. In: Fifth Conference on Uncertainty in Artificial Intelligence, pp. 139–148 (1989)
3. Kjærulff, U.B., Madsen, A.L.: Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis, 2nd edn. Springer, New York (2013)
4. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press, Cambridge (2009)
5. Lauritzen, S.L., Dawid, A.P., Larsen, B.N., Leimer, H.G.: Independence properties of directed Markov fields. *Networks* **20**, 491–505 (1990)
6. Lauritzen, S.L., Spiegelhalter, D.J.: Local computation with probabilities on graphical structures and their application to expert systems. *J. Roy. Stat. Soc.* **50**, 157–244 (1988)
7. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, Burlington (1988)
8. Pearl, J.: Fusion, propagation and structuring in belief networks. *Artif. Intell.* **29**, 241–288 (1986)
9. Pearl, J.: Belief networks revisited. *Artif. Intell.* **59**, 49–56 (1993)
10. Pearl, J.: Causality. Cambridge University Press, Cambridge (2009)
11. Verma, T., Pearl, J.: Equivalence and synthesis of causal models. In: Sixth Conference on Uncertainty in Artificial Intelligence, pp. 220–227. GE Corporate Research and Development (1990)
12. Wong, S.K.M., Butz, C.J., Wu, D.: On the implication problem for probabilistic conditional independency. *IEEE Trans. Syst. Man Cybern. Part A: Syst. Humans* **30**(6), 785–805 (2000)
13. Zhang, N.L., Poole, D.: A simple approach to Bayesian network computations. In: Proceedings of the Tenth Canadian Artificial Intelligence Conference, pp. 171–178 (1994)