Algorithm 3.1 Algorithm for finding nodes reachable from X given $\boldsymbol{Z}$ via active trails

```
    Procedure Reachable (
      G,    // Bayesian network graph
      X,    // Source variable
      Z     // Observations
    )
1     // Phase I: Insert all ancestors of Z into A
2   L ← Z     // Nodes to be visited
3   A ← ∅     // Ancestors of Z
4   while L ≠ ∅
5     Select some Y from L
6     L ← L − {Y}
7     if Y ∉ A then
8       L ← L ∪ Pa_Y     // Y's parents need to be visited
9     A ← A ∪ {Y}     // Y is ancestor of evidence
10
11    // Phase II: traverse active trails starting from X
12  L ← {(X, ↑)}     // (Node,direction) to be visited
13  V ← ∅     // (Node,direction) marked as visited
14  R ← ∅     // Nodes reachable via active trail
15  while L ≠ ∅
16    Select some (Y, d) from L
17    L ← L − {(Y, d)}
18    if (Y, d) ∉ V then
19      if Y ∉ Z then
20        R ← R ∪ {Y}     // Y is reachable
21      V ← V ∪ {(Y, d)}     // Mark (Y, d) as visited
22      if d =↑ and Y ∉ Z then     // Trail up through Y active if Y not in Z
23        for each Z ∈ Pa_Y
24          L ← L ∪ {(Z, ↑)}     // Y's parents to be visited from bottom
25        for each Z ∈ Ch_Y
26          L ← L ∪ {(Z, ↓)}     // Y's children to be visited from top
27      else if d =↓ then     // Trails down through Y
28        if Y ∉ Z then
29          // Downward trails to Y's children are active
30        for each Z ∈ Ch_Y
31          L ← L∪{(Z, ↓)}     // Y's children to be visited from top
32        if Y ∈ A then     // v-structure trails are active
33          for each Z ∈ Pa_Y
34            L ← L∪{(Z, ↑)}     // Y's parents to be visited from bottom
35  return R
```