CompSci 171: Intro AI

Homework 6

Propositional Logic

1

Statement:

- The car is either at John's house or at Fred's house.
- If the car is not at John's house, then it must be at Fred's house

1 (a)

Set of propositional letters which can be used to represent this statement:

X: car is at John’s house

Y: car is at Fred’s house

¬X: car is not at John’s house

1 (b)

- The car is either at John's house or at Fred's house:
  - X V Y
  - (X Λ ¬Y) V (¬X Λ Y) (car cannot be at John's house and at Fred's house at the same time)
- If the car is not at John's house, then it must be at Fred's house
  - ¬X => Y

1 (c)

- Can we determine where the car is?

X	Y	X∧¬Y	¬X∧Y	(X∧¬Y)∨(¬X∧Y)	¬X=>Y	(X∧¬Y)∨(¬X∧Y)) ∧ (¬X => Y)
T	T	F	F	F	T	F
T	F	T	F	T	T	T
F	T	F	T	T	T	T
F	F	F	F	F	F	T

No – the car can be either at John's or at Fred's house

7.4 (a)

- α is valid if and only if True $\models \alpha$

A sentence is valid if it is true in all models.

α
True

True entails α if and only if for every model in which True is true α is also true

True	α
True	True

7.4 (c)

- ($\alpha \models \beta$) if and only if the sentence ($\alpha \Rightarrow \beta$) is valid

$\alpha \models \beta$

α	β
True	True
False	True
False	False

$\alpha \Rightarrow \beta$ is valid

α	β	$\alpha \Rightarrow \beta$
True	True	True
False	True	True
False	False	True

7.4 (e)

- ($\alpha \models \beta$) if and only if the sentence ($\alpha \wedge \neg \beta$) is unsatisfiable

$\alpha \models \beta$

α	β
True	True
False	True
False	False

$\alpha \wedge \neg \beta$ is unsatisfiable

α	β	$\alpha \wedge \neg \beta$
True	True	False
False	True	False
False	False	False

7.5

"*m*" is a model of α means that sentence α is true in model *m*

7.5 (a)

(A ^ B) v (B ^ C): 3 models

A	B	C	A^B	B^C	(A ^ B) v (B ^ C)
F	F	F	F	F	F
F	F	T	F	F	F
F	T	F	F	F	F
F	T	T	F	T	T
T	F	F	F	F	F
T	F	T	F	F	F
T	T	F	T	F	T
T	T	T	T	T	T

7.5 b

A v B: 3 models

A	B	A v B
F	F	F
F	T	T
T	F	T
T	T	T

7.8 c

(Smoke => Fire) => (¬Smoke => ¬Fire)

We know : (a => b) ≡ (¬a v b):

(Smoke => Fire) => (¬Smoke => ¬Fire) ≡ (¬ (Smoke => Fire) v (¬Smoke => ¬Fire))
(Smoke => Fire) ≡ (¬Smoke v Fire)
(¬Smoke => ¬Fire) ≡ (Smoke v ¬ Fire)

(Smoke => Fire) => (¬Smoke => ¬Fire) ≡ ((¬ (¬Smoke v Fire) v (Smoke v ¬ Fire))

We know: ¬ (a v b) ≡ (¬a ^ ¬b):

(Smoke => Fire) => (¬Smoke => ¬Fire) ≡
(Smoke ^ ¬ Fire) v (Smoke v ¬ Fire) ≡
(Smoke v Smoke v¬ Fire) ^ (¬ Fire v Smoke v¬ Fire) ≡
(Smoke v ¬ Fire) ^(Smoke v ¬ Fire) ≡(Smoke v ¬ Fire)

NEITHER

7.8 d

Smoke v Fire v ¬Fire

Smoke	Fire	¬Fire	Fire v ¬Fire	Smoke v Fire v ¬Fire
F	F	T	T	T
F	T	F	T	T
T	F	T	T	T
T	T	F	T	T

VALID

7.8 f

(Smoke => Fire) => ((Smoke ^ Heat) => Fire)

We know : (a => b) ≡ (¬a v b):

(Smoke => Fire) => ((Smoke ^ Heat) => Fire) ≡

(¬ (Smoke => Fire) v ((Smoke ^ Heat) => Fire) ≡

(¬ (¬ Smoke v Fire) v (¬ (Smoke ^ Heat) v Fire)) ≡

(Smoke ^ ¬ Fire) v (¬ Smoke v ¬ Heat v Fire) ≡

(Smoke v ¬ Smoke v ¬ Heat v Fire) ^ (¬ Fire v ¬ Smoke v ¬ Heat v Fire) ≡

((Smoke v ¬ Smoke) v ¬ Heat v Fire) ^ ((¬ Fire v Fire) v¬ Smoke v ¬ Heat) ≡

(True v ¬ Heat v Fire) ^ (True v¬ Smoke v ¬ Heat) ≡ True ^ True ≡ True

VALID

7.8 h

(Big ^ Dumb) v ¬ Dumb

Big	Dumb	¬ Dumb	Big ^ Dumb	(Big ^ Dumb) v ¬ Dumb
F	F	T	F	T
F	T	F	F	F
T	F	T	F	T
T	T	F	T	T

NEITHER

1. ¬[((P v ¬Q) → R) → (P ^ R)]

¬[((P v ¬Q) → R) → (P ^ R)] ≡ Implication elimination: (a → b) ≡ (¬a v b)

1. ¬[((P v ¬Q) → R) → (P ^ R)]

¬[((P v ¬Q) → R) → (P ^ R)] ≡ Implication elimination: (a → b) ≡ (¬a v b)

¬[¬((P v ¬Q) → R) v (P ^ R)] ≡ Implication elimination: (a → b) ≡ (¬a v b)

1. ¬[((P v ¬Q) → R) → (P ^ R)]

¬[((P v ¬Q) → R) → (P ^ R)] ≡ Implication elimination: (a → b) ≡ (¬a v b)

¬[¬((P v ¬Q) → R) v (P ^ R)] ≡ Implication elimination: (a → b) ≡ (¬a v b)

¬[¬(¬(P v ¬Q) v R) v (P ^ R)] ≡ De Morgan: ¬ (a v b) ≡ (¬a ^ ¬b)

1. ¬[((P v ¬Q) → R) → (P ^ R)]

¬[((P v ¬Q) → R) → (P ^ R)] ≡ Implication elimination: (a → b) ≡ (¬a ^ b)

¬[¬((P v ¬Q) → R) v (P ^ R)] ≡ Implication elimination: (a → b) ≡ (¬a ^ b)

¬[¬(¬(P v ¬Q) v R) v (P ^ R)] ≡ De Morgan: ¬ (a v b) ≡ (¬a ^ ¬b)

¬[¬((¬P ^ Q) v R) v (P ^ R)] ≡ De Morgan: ¬ (a v b) ≡ (¬a ^ ¬b)

1. ¬[((P v ¬Q) → R) → (P ^ R)]

¬[((P v ¬Q) → R) → (P ^ R)] ≡ Implication elimination: (a → b) ≡ (¬a ^ b)

¬[¬((P v ¬Q) → R) v (P ^ R)] ≡ Implication elimination: (a → b) ≡ (¬a ^ b)

¬[¬(¬(P v ¬Q) v R) v (P ^ R)] ≡ De Morgan: ¬ (a v b) ≡ (¬a ^ ¬b)

¬[¬((¬P ^ Q) v R) v (P ^ R)] ≡ De Morgan: ¬ (a v b) ≡ (¬a ^ ¬b)

[((¬P ^ Q) v R) ^ ¬ (P ^ R)] ≡ De Morgan: ¬ (a ^ b) ≡ (¬a v ¬b)

1. ¬[((P v ¬Q) → R) → (P ^ R)]

¬[((P v ¬Q) → R) → (P ^ R)] ≡ Implication elimination: (a → b) ≡ (¬a ^ b)

¬[¬((P v ¬Q) → R) v (P ^ R)] ≡ Implication elimination: (a → b) ≡ (¬a ^ b)

¬[¬(¬(P v ¬Q) v R) v (P ^ R)] ≡ De Morgan: ¬ (a v b) ≡ (¬a ^ ¬b)

¬[¬((¬P ^ Q) v R) v (P ^ R)] ≡ De Morgan: ¬ (a v b) ≡ (¬a ^ ¬b)

[((¬P ^ Q) v R) ^ ¬ (P ^ R)] ≡ De Morgan: ¬ (a ^ b) ≡ (¬a v ¬b)

[((¬P ^ Q) v R) ^ (¬P v ¬R)] ≡ Distributivity of v over ^:
(a ^ b) v c ≡ ((a v c) ^ (b v c)

1. ¬[((P v ¬Q) → R) → (P ^ R)]

¬[((P v ¬Q) → R) → (P ^ R)] ≡ Implication elimination: (a → b) ≡ (¬a ^ b)

¬[¬((P v ¬Q) → R) v (P ^ R)] ≡ Implication elimination: (a → b) ≡ (¬a ^ b)

¬[¬(¬(P v ¬Q) v R) v (P ^ R)] ≡ De Morgan: ¬ (a v b) ≡ (¬a ^ ¬b)

¬[¬((¬P ^ Q) v R) v (P ^ R)] ≡ De Morgan: ¬ (a v b) ≡ (¬a ^ ¬b)

[((¬P ^ Q) v R) ^ ¬ (P ^ R)] ≡ De Morgan: ¬ (a ^ b) ≡ (¬a v ¬b)

[((¬P ^ Q) v R) ^ (¬P v ¬R)] ≡ Distributivity of v over ^:
(a ^ b) v c ≡ ((a v c) ^ (b v c)

[(¬P v R) ^ (Q v R) ^ (¬P v ¬R)] ≡

(¬P v R) ^ (Q v R) ^ (¬P v ¬R)

2. Use resolution Algorithm to solve the following problem

Given (KB):

$B \wedge C \rightarrow A$

B

$D \wedge E \rightarrow C$

$E \vee F$

$D \wedge \neg F$

Query:

A

$KB \models A$ iff $(KB \wedge \neg A)$ is unsatisfiable

2. Can we entail the query from the knowledge base

Resolution rule:

$$\frac{a \vee b, \neg b \vee c}{a \vee c}$$

New clause contains all the literals of two original clauses except the two complementary literals (b and ¬b)

And-Elimination:

$$\frac{a \wedge b}{a} \qquad \frac{a \wedge b}{b}$$

From a conjunction any of the conjuncts can be inferred

2. Use resolution Algorithm to solve the following problem

Given (KB):

$B \wedge C \rightarrow A \equiv (\neg(B \wedge C) \vee A) \equiv (\neg B \vee \neg C \vee A)$

B

$D \wedge E \rightarrow C \equiv (\neg(D \wedge E) \vee C) \equiv (\neg D \vee \neg E \vee C)$

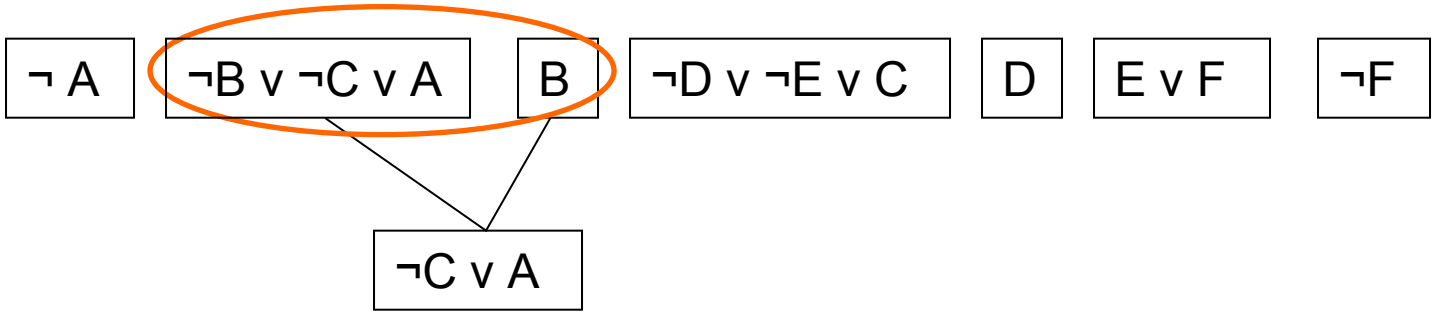
$E \vee F$

$D \wedge \neg F$

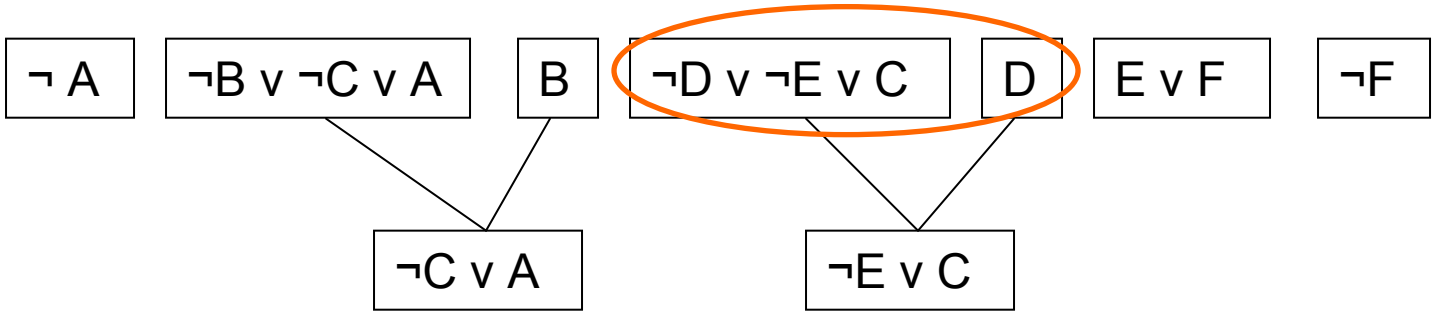
2. Use resolution Algorithm to solve the following problem

¬ A	¬B v ¬C v A	B	¬D v ¬E v C	D	E v F	¬F

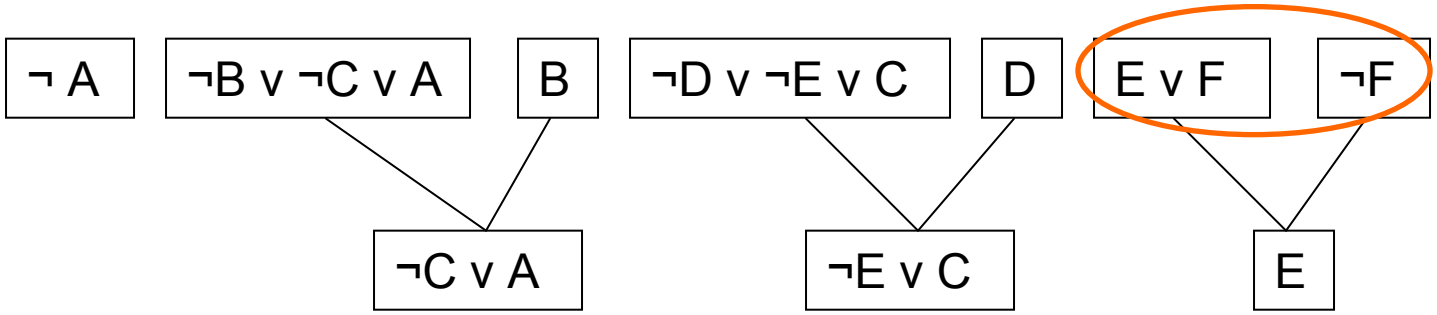
2. Use resolution Algorithm to solve the following problem



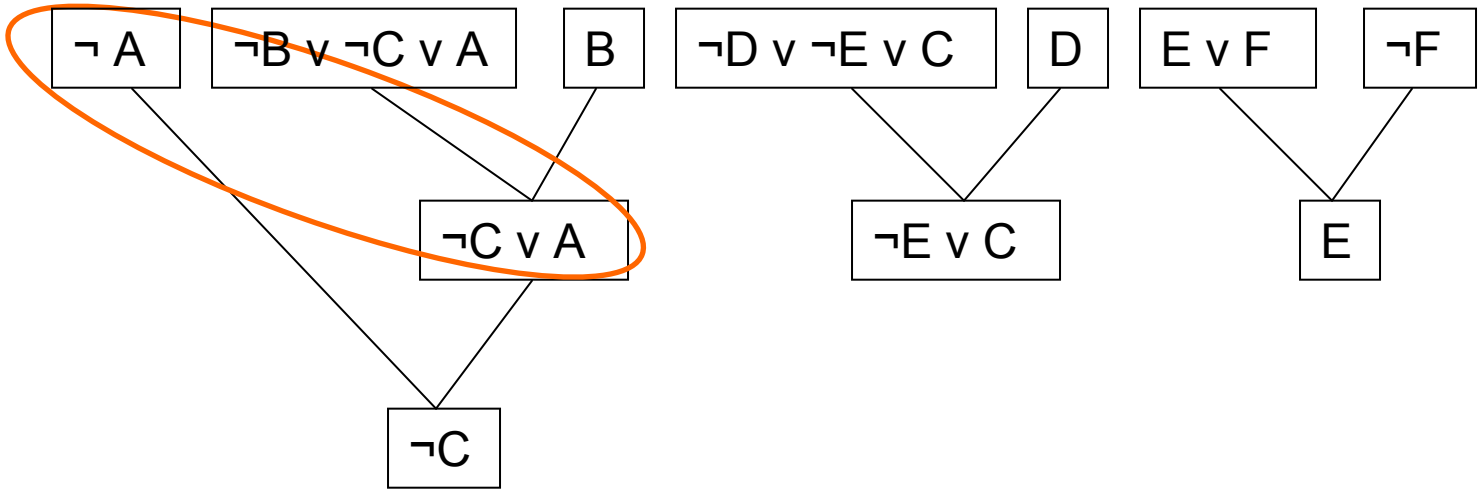
2. Use resolution Algorithm to solve the following problem



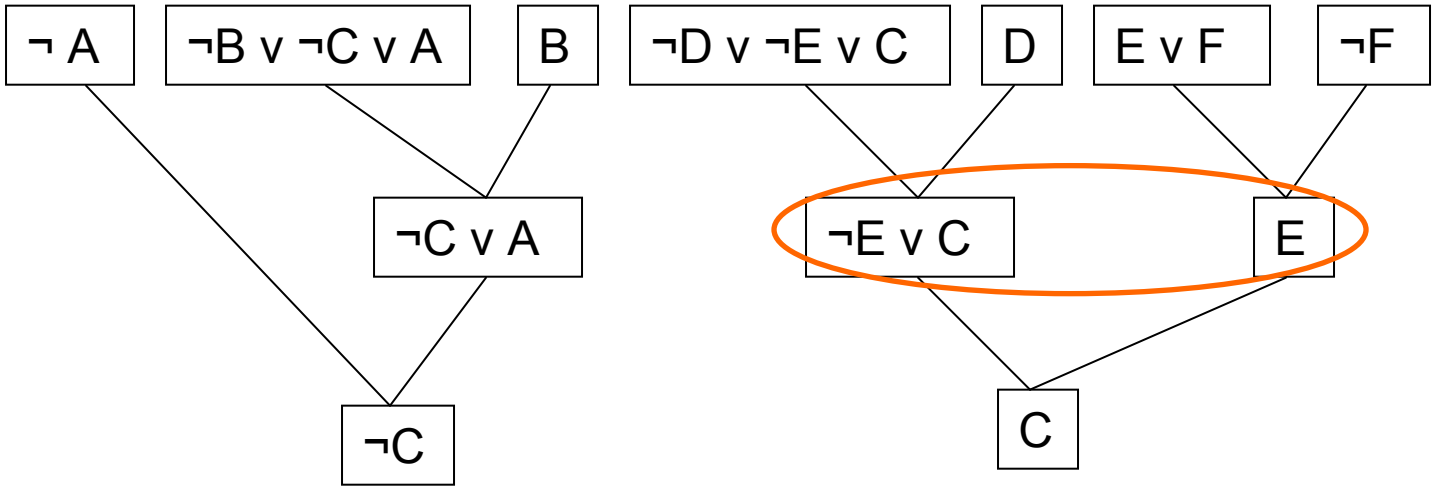
2. Use resolution Algorithm to solve the following problem



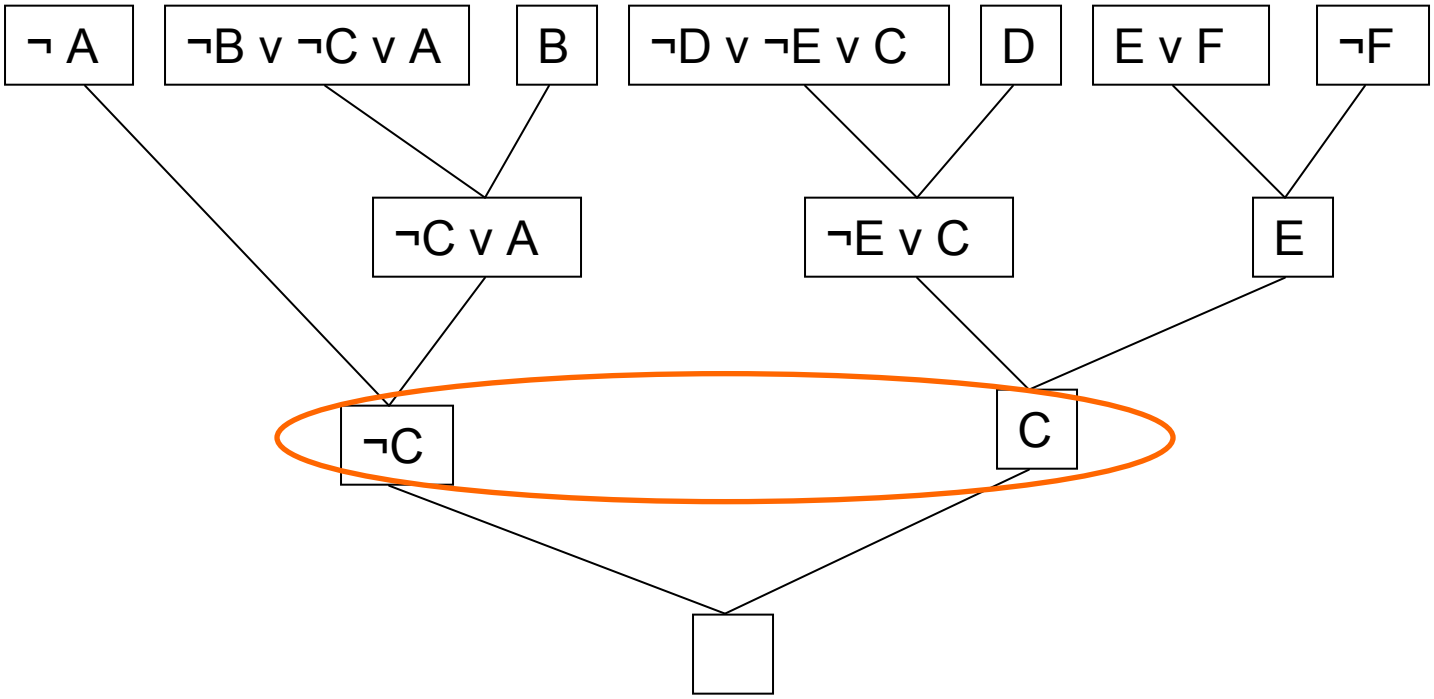
2. Use resolution Algorithm to solve the following problem



2. Use resolution Algorithm to solve the following problem



2. Use resolution Algorithm to solve the following problem



Thus, we can entail query A from the knowledge base KB

3. Use Forward Chaining to solve the problem

Given:

A

B

C

$A \wedge B \rightarrow D$

$B \wedge D \rightarrow F$

$F \rightarrow G$

$A \wedge E \rightarrow H$

$A \wedge C \rightarrow E$

Is H true?

3. Use Forward Chaining to solve the problem

Given:

A

B

C

A ^ B → D

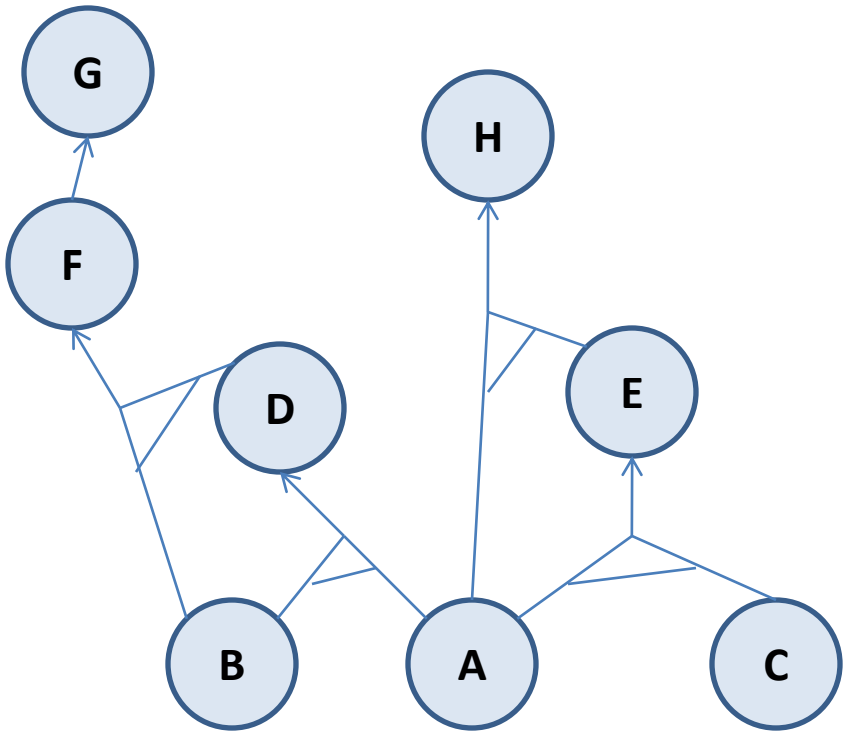
B ^ D → F

F → G

A ^ E → H

A ^ C → E

Is H true?



3. Use Forward Chaining to solve the problem

Given:

A

B

C

A ^ B → D

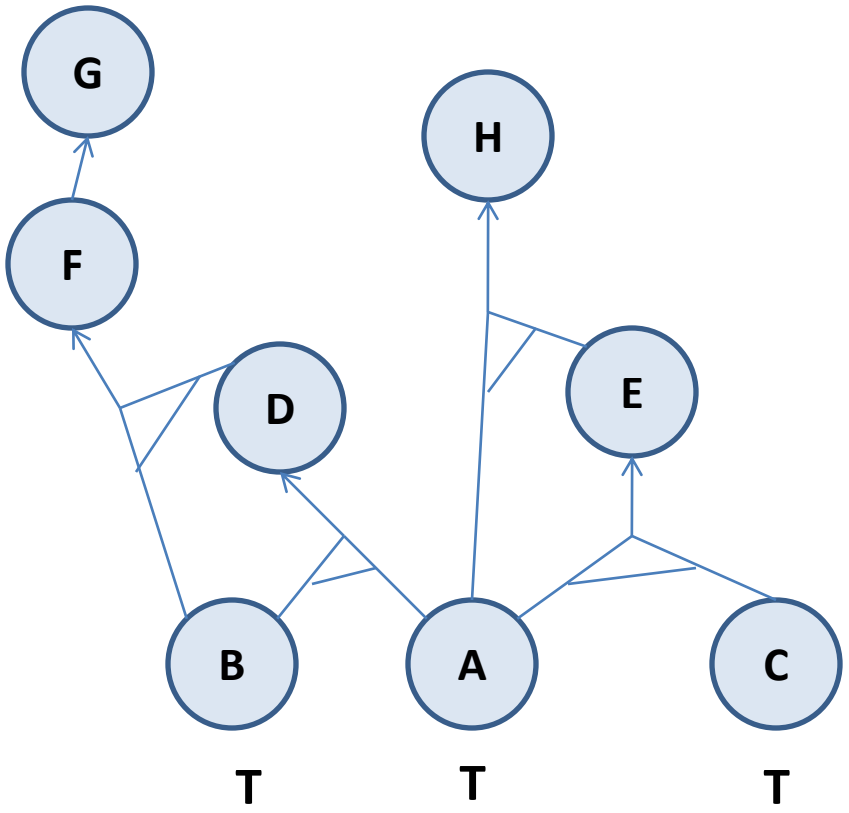
B ^ D → F

F → G

A ^ E → H

A ^ C → E

Is H true?



3. Use Forward Chaining to solve the problem

Given:

A

B

C

A ^ B → D

B ^ D → F

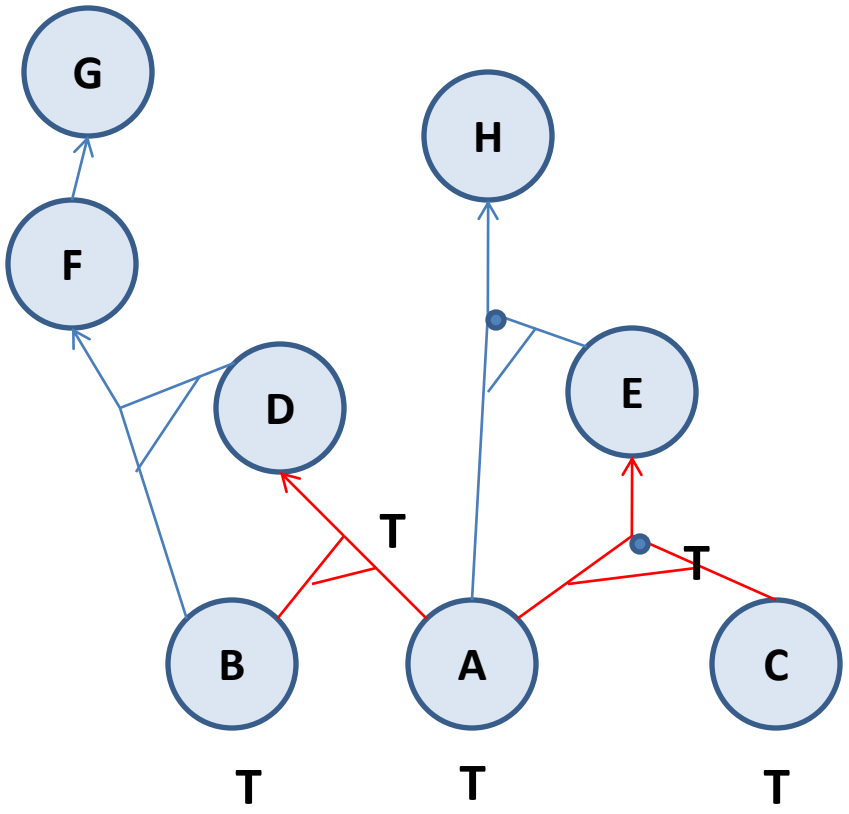
F → G

A ^ E → H

A ^ C → E

A ^ C

A ^ B



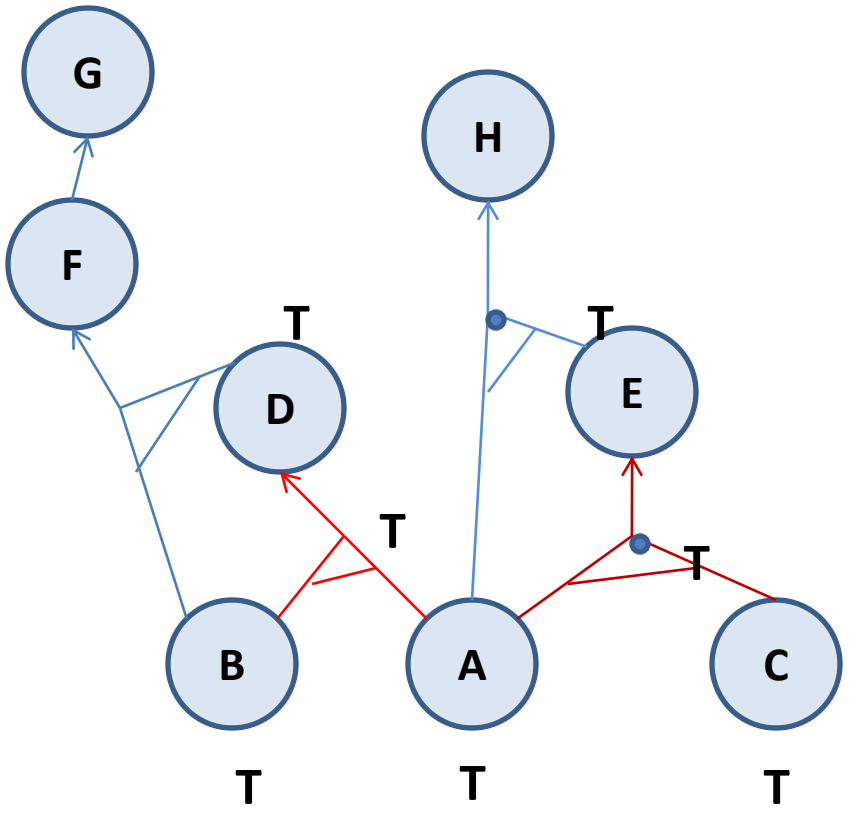
3. Use Forward Chaining to solve the problem

Given:

A
B
C
A ^ B → D
B ^ D → F
F → G
A ^ E → H
A ^ C → E

A ^ C
A ^ B

E
D



3. Use Forward Chaining to solve the problem

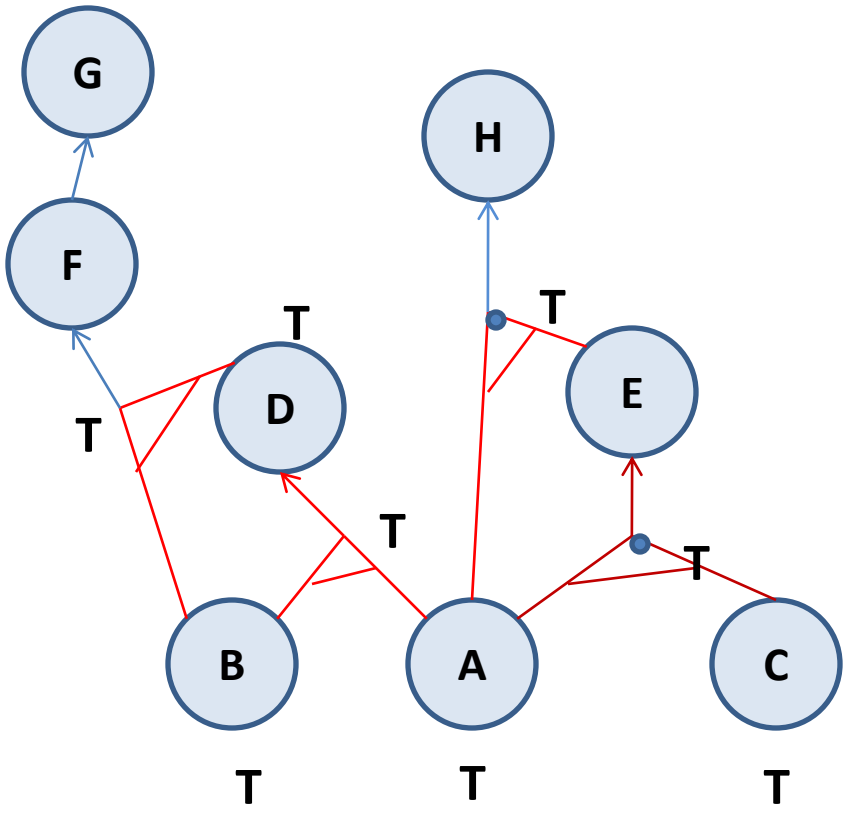
Given:

A
B
C
A ^ B → D
B ^ D → F
F → G
A ^ E → H
A ^ C → E

A ^ C
A ^ B

E
D

A ^ E
B ^ E



3. Use Forward Chaining to solve the problem

Given:

A
B
C
A ^ B → D
B ^ D → F
F → G
A ^ E → H
A ^ C → E

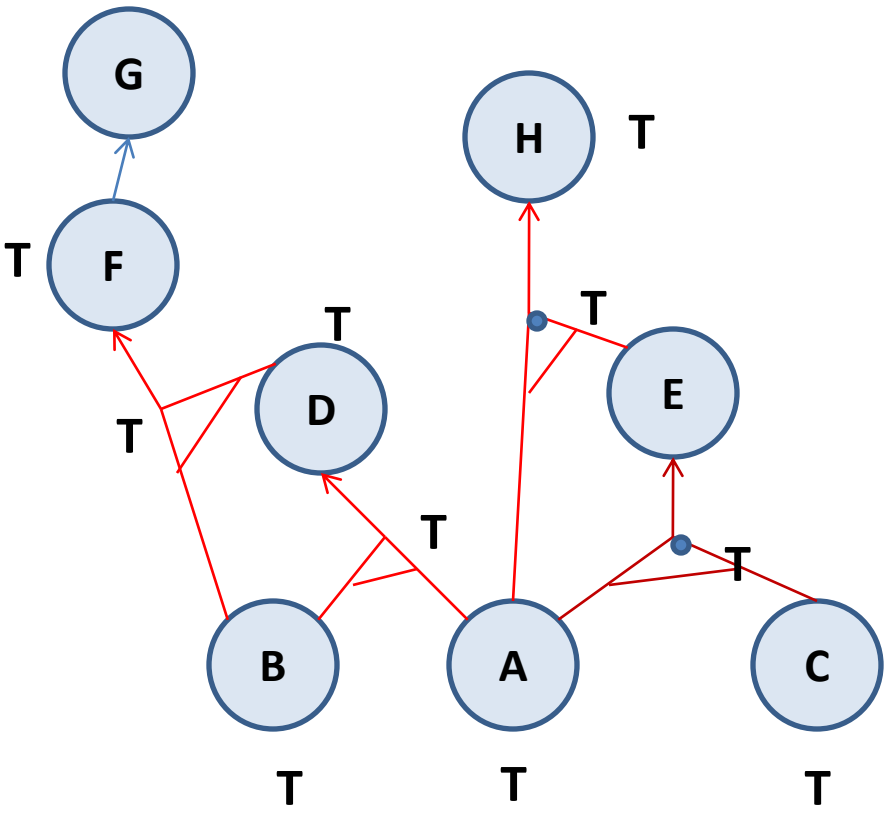
A ^ C
A ^ B

E
D

A ^ E
B ^ E

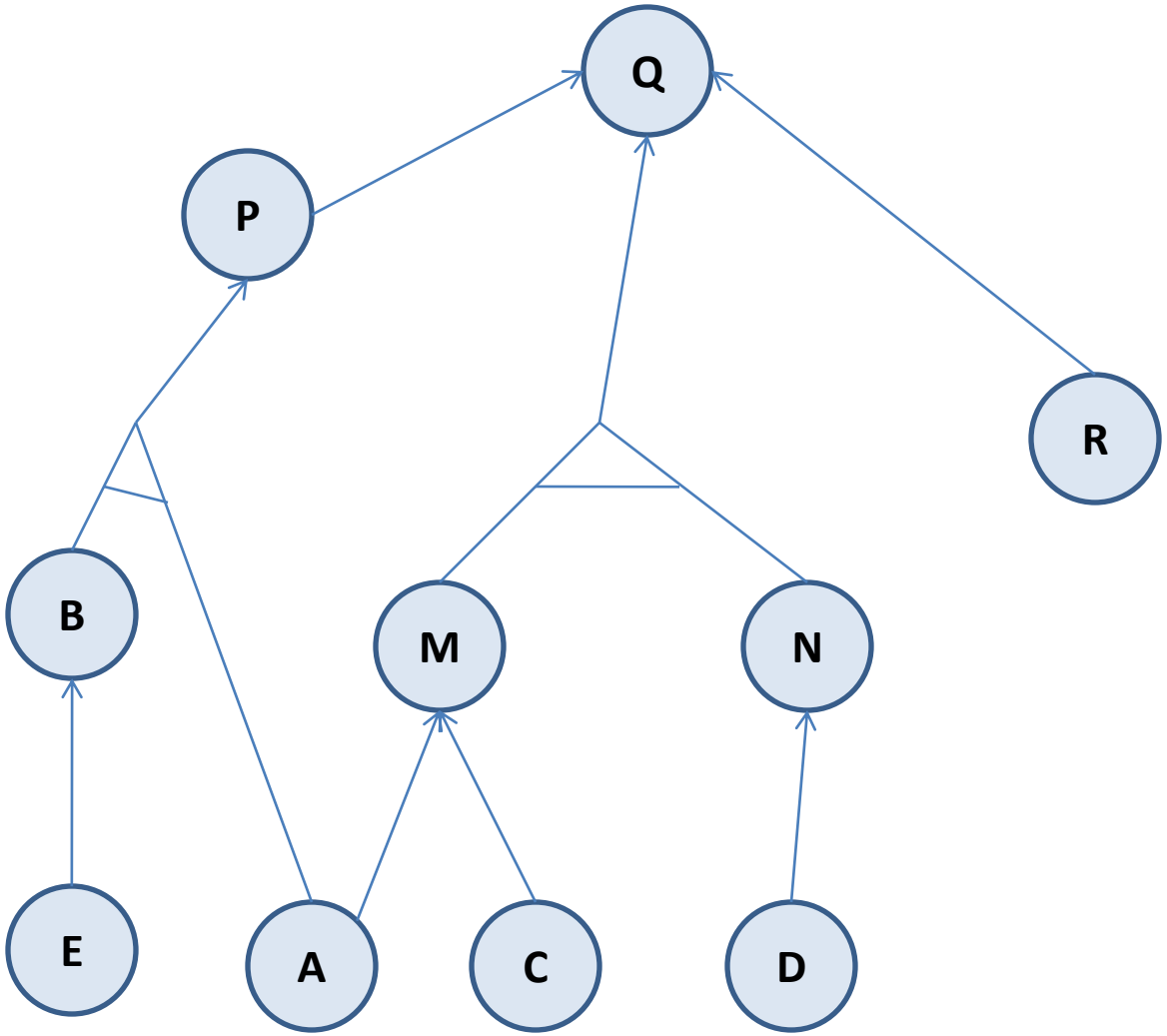
H
F

H is true!



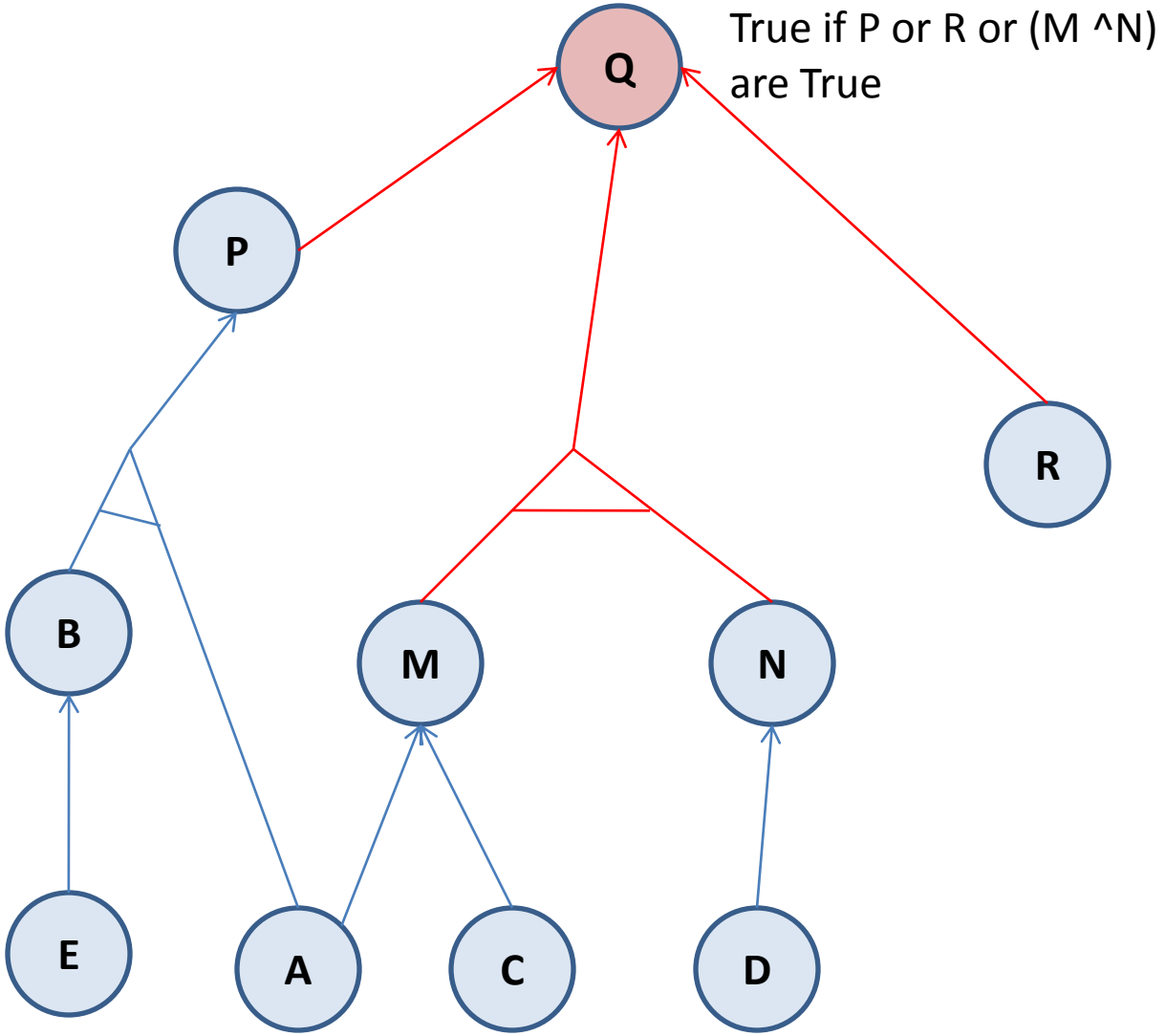
4. Use backward Chaining to prove Q

P → Q
E → B
R → Q
M ^ N → Q
A ^ B → P
A → M
C → M
D → N
D
A



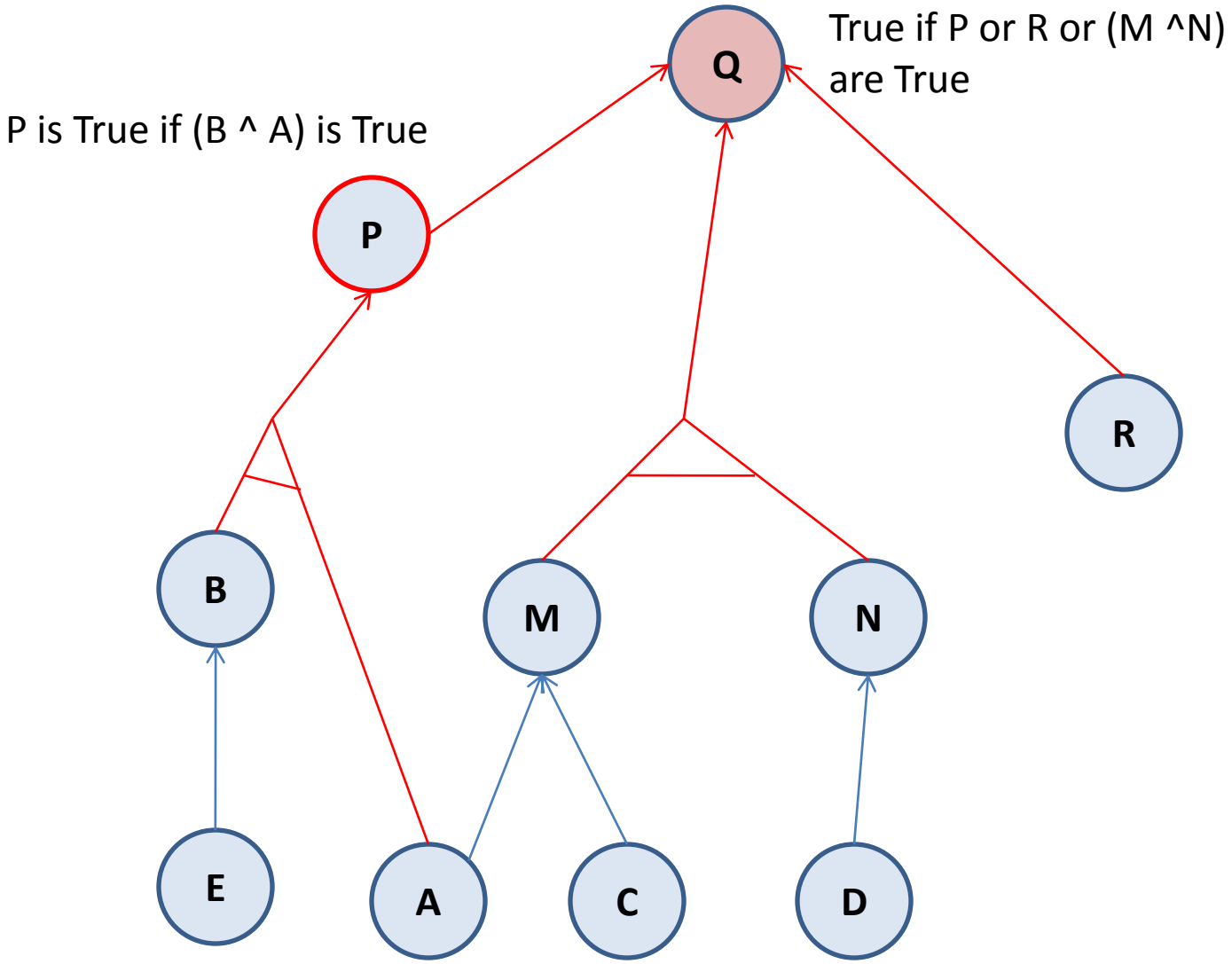
4. Use backward Chaining to prove Q

P → Q
E → B
R → Q
M ^ N → Q
A ^ B → P
A → M
C → M
D → N
D
A



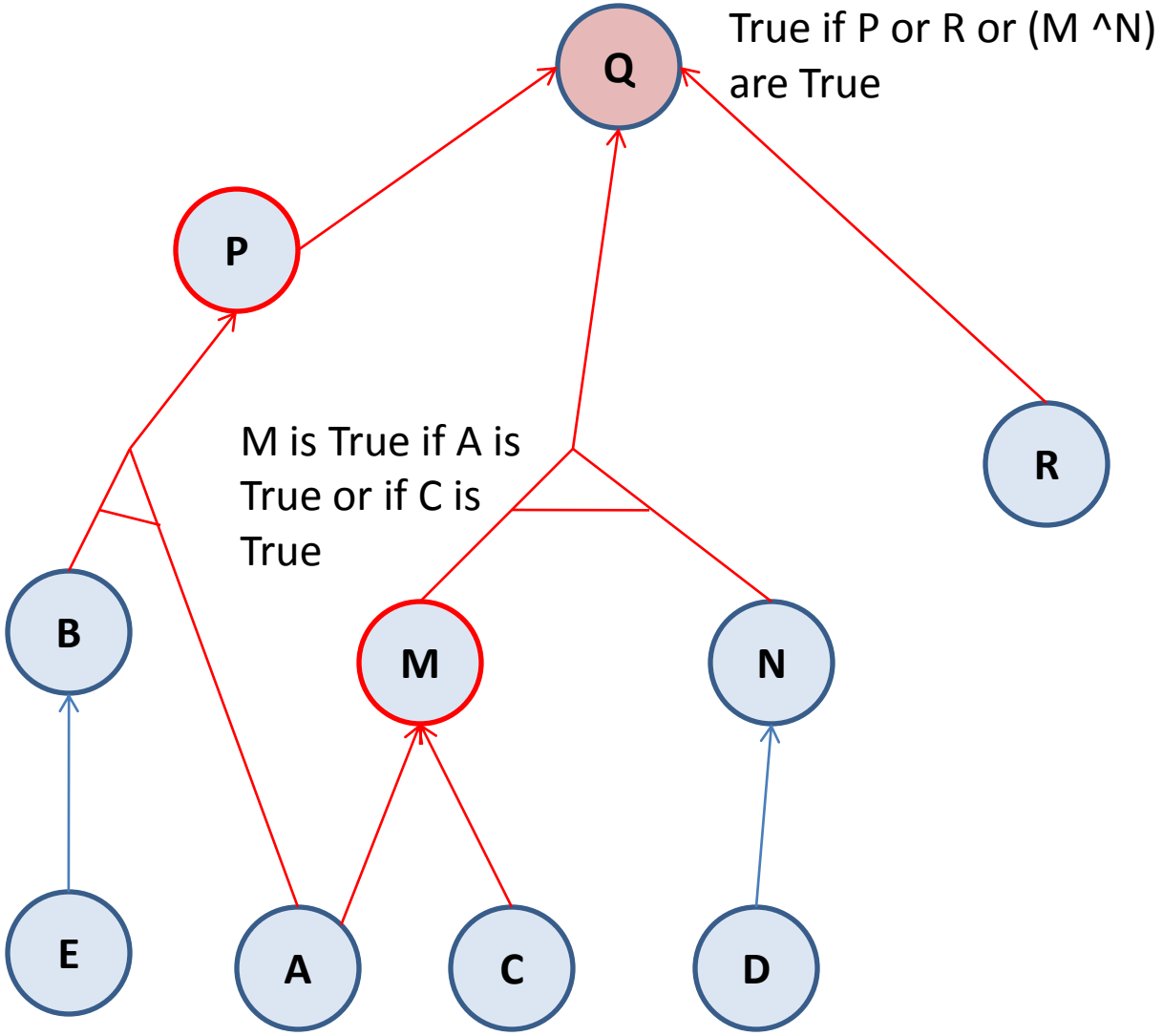
4. Use backward Chaining to prove Q

P → Q
E → B
R → Q
M ^ N → Q
A ^ B → P
A → M
C → M
D → N
D
A



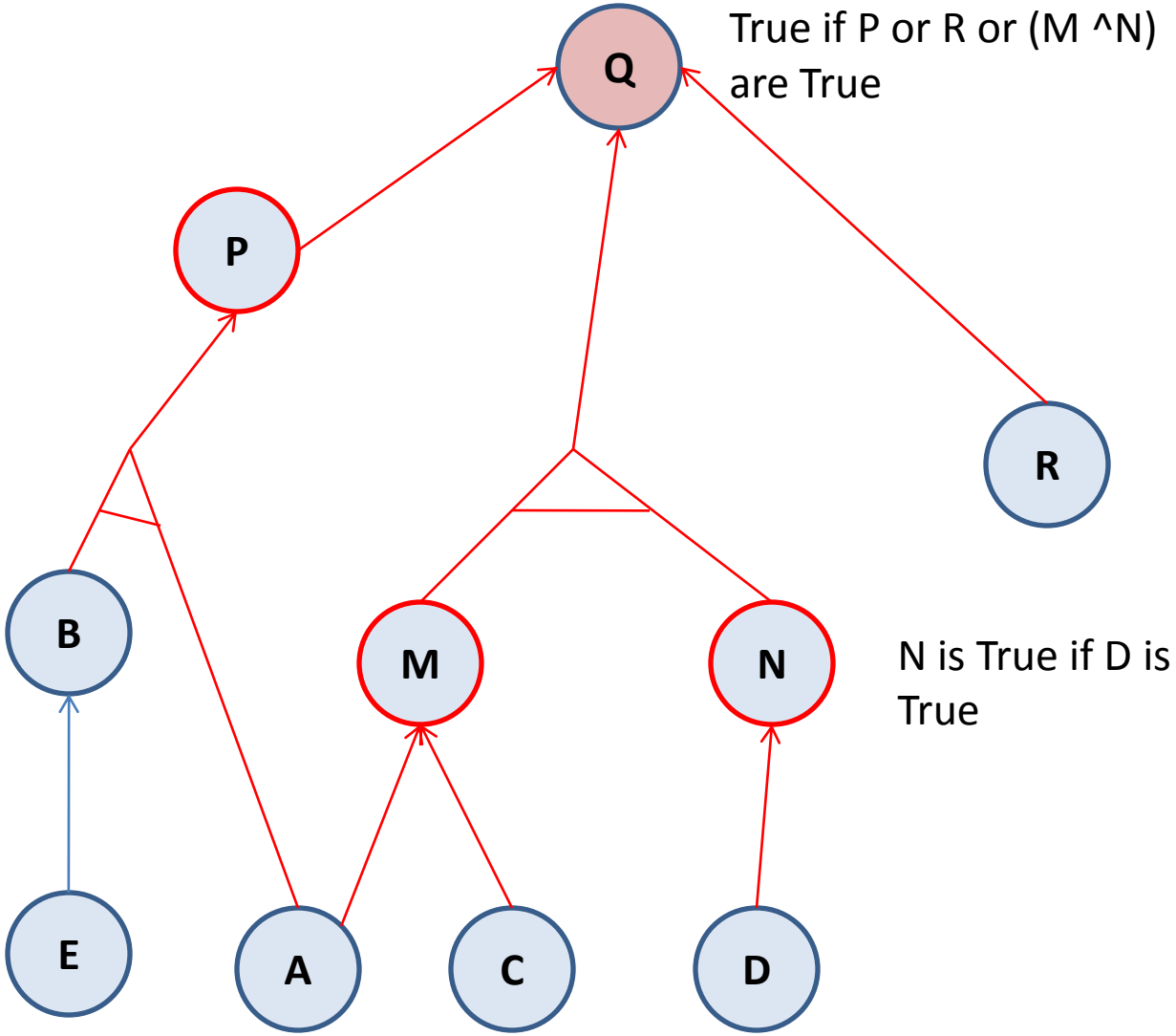
4. Use backward Chaining to prove Q

P → Q
E → B
R → Q
M ^ N → Q
A ^ B → P
A → M
C → M
D → N
D
A



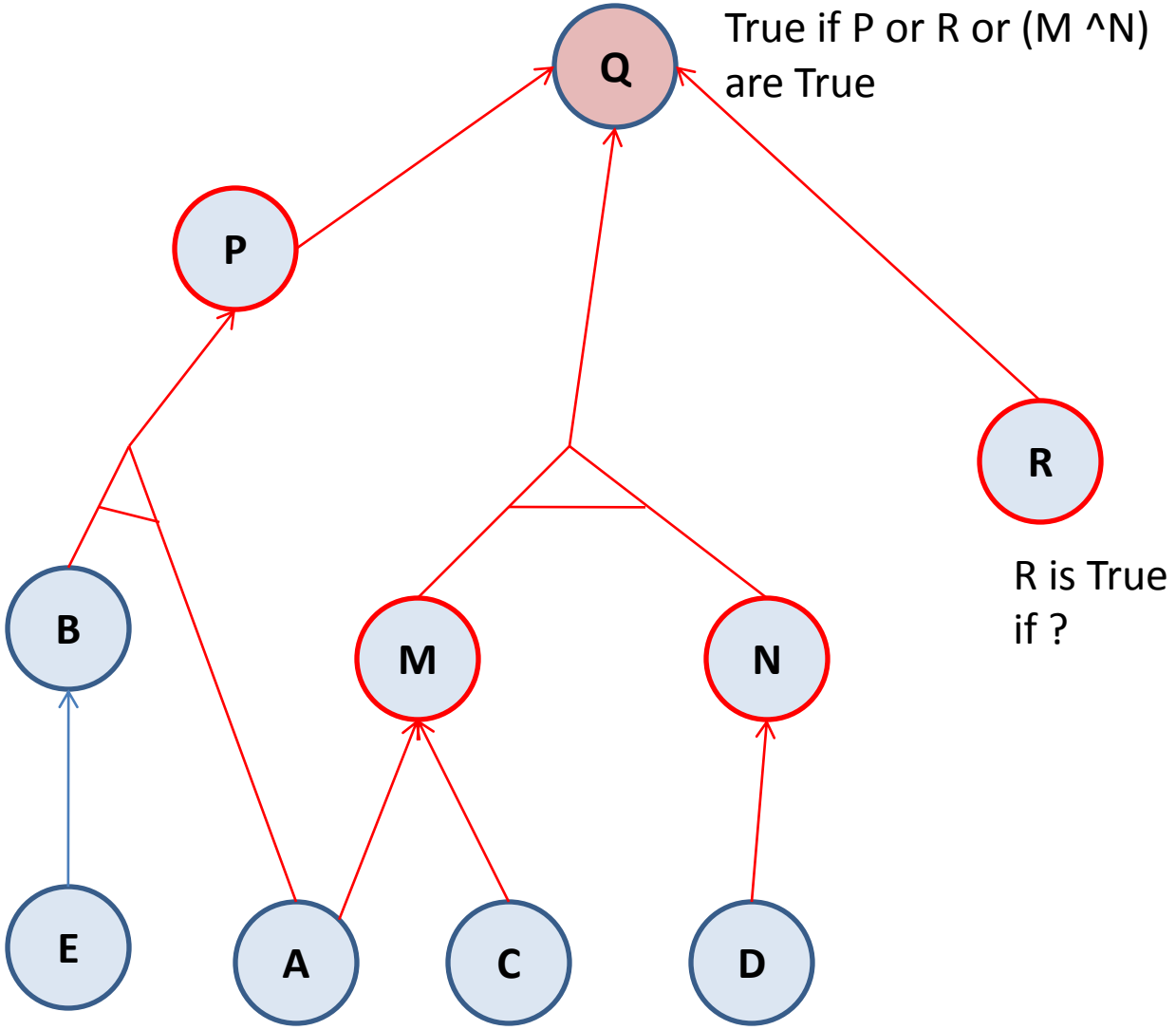
4. Use backward Chaining to prove Q

P → Q
E → B
R → Q
M ^ N → Q
A ^ B → P
A → M
C → M
D → N
D
A



4. Use backward Chaining to prove Q

P → Q
E → B
R → Q
M ^ N → Q
A ^ B → P
A → M
C → M
D → N
D
A



4. Use backward Chaining to prove Q

$P \rightarrow Q$

$E \rightarrow B$

$R \rightarrow Q$

$M \wedge N \rightarrow Q$

$A \wedge B \rightarrow P$

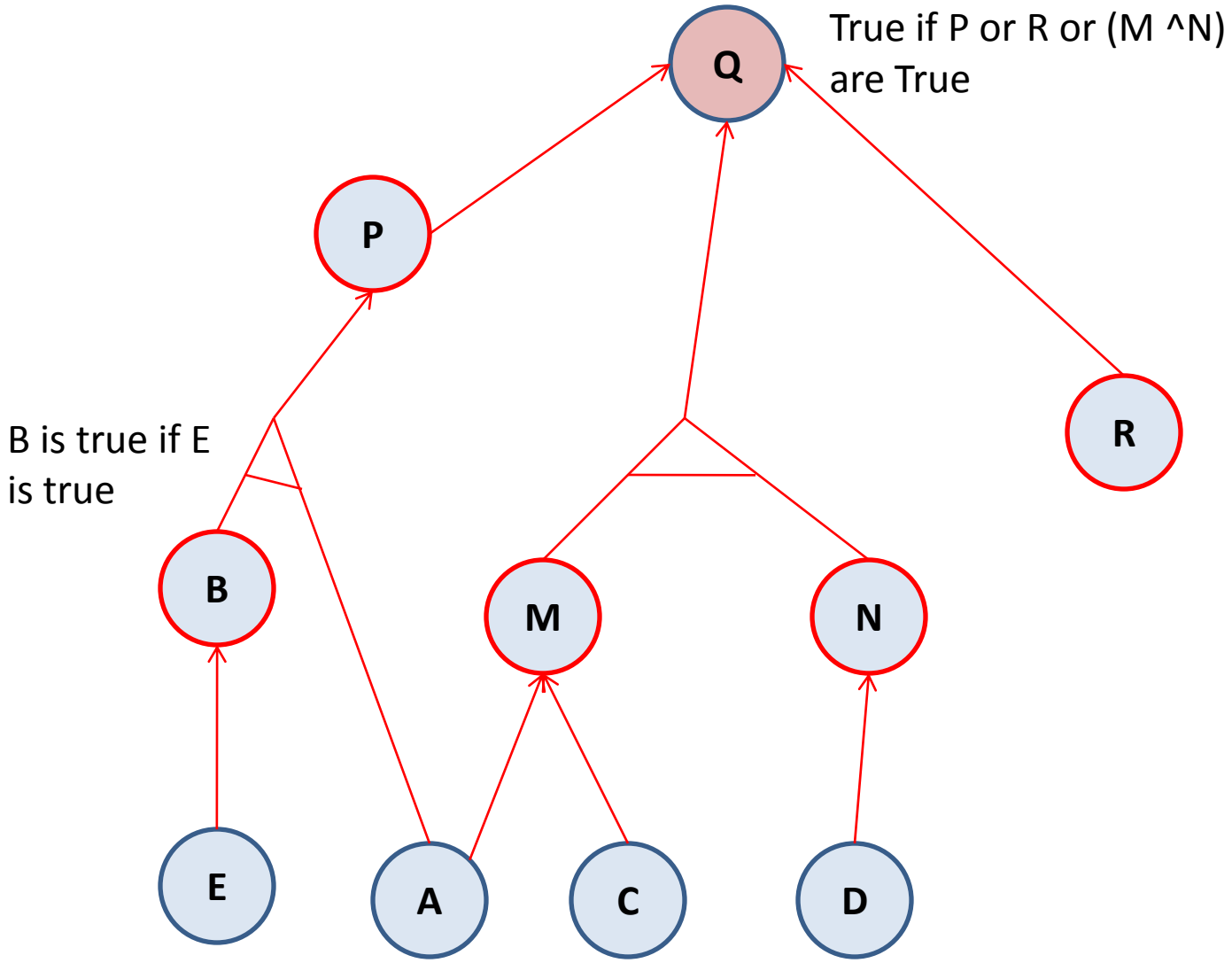
$A \rightarrow M$

$C \rightarrow M$

$D \rightarrow N$

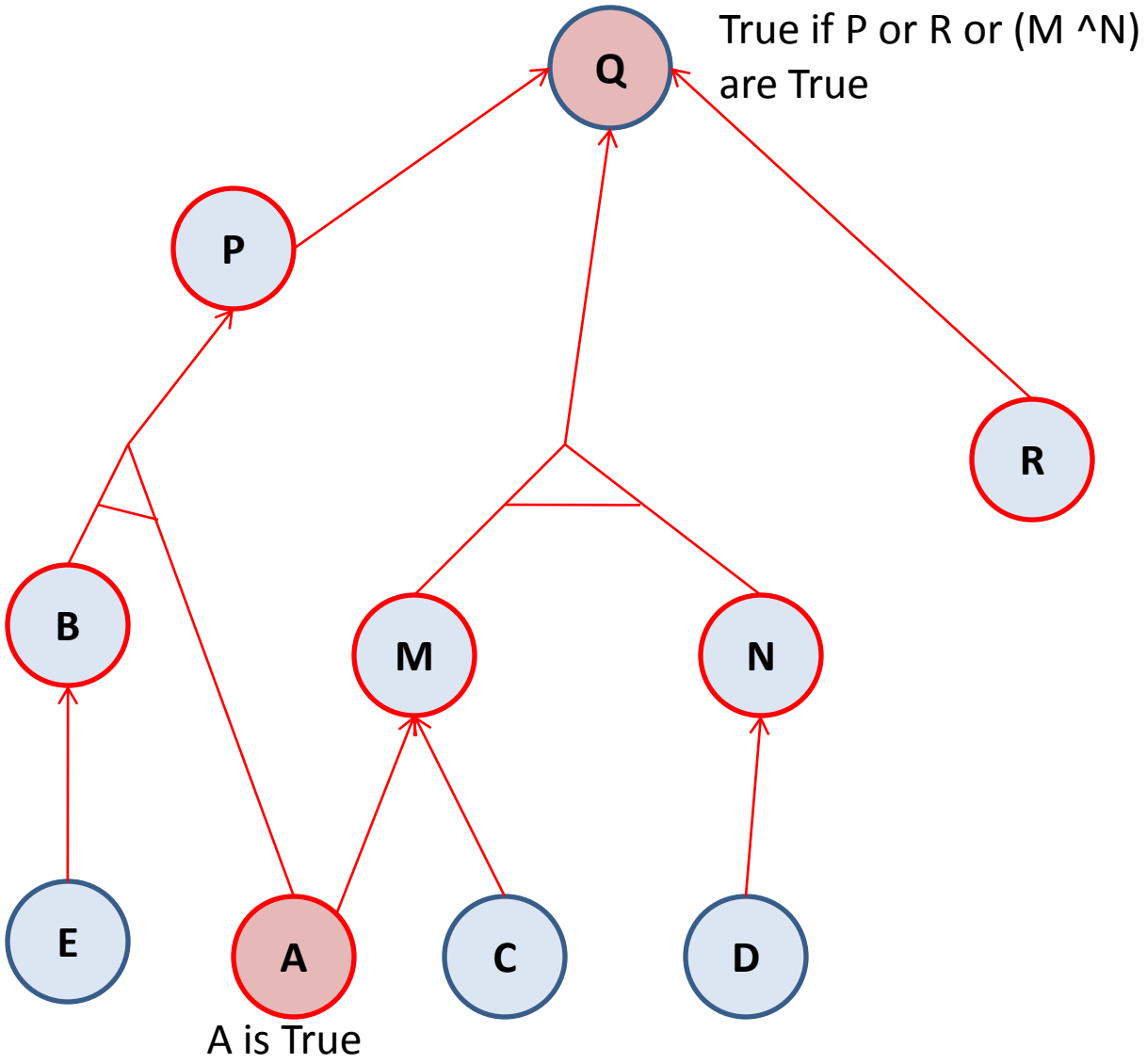
D

A



4. Use backward Chaining to prove Q

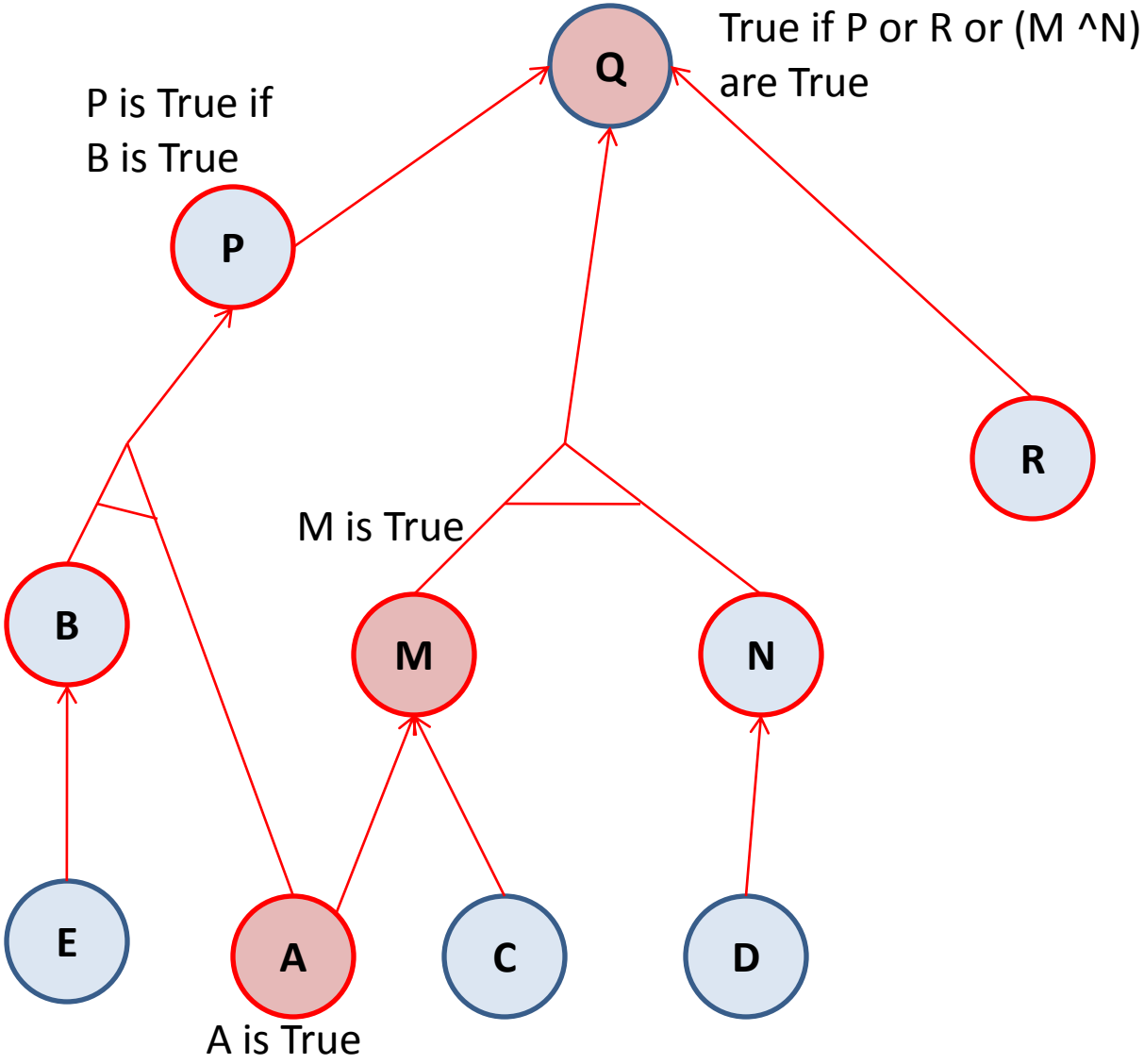
P → Q
E → B
R → Q
M ^ N → Q
A ^ B → P
A → M
C → M
D → N
D
A



4. Use backward Chaining to prove Q

P → Q
E → B
R → Q
M ^ N → Q
A ^ B → P
A → M
C → M
D → N
D
A

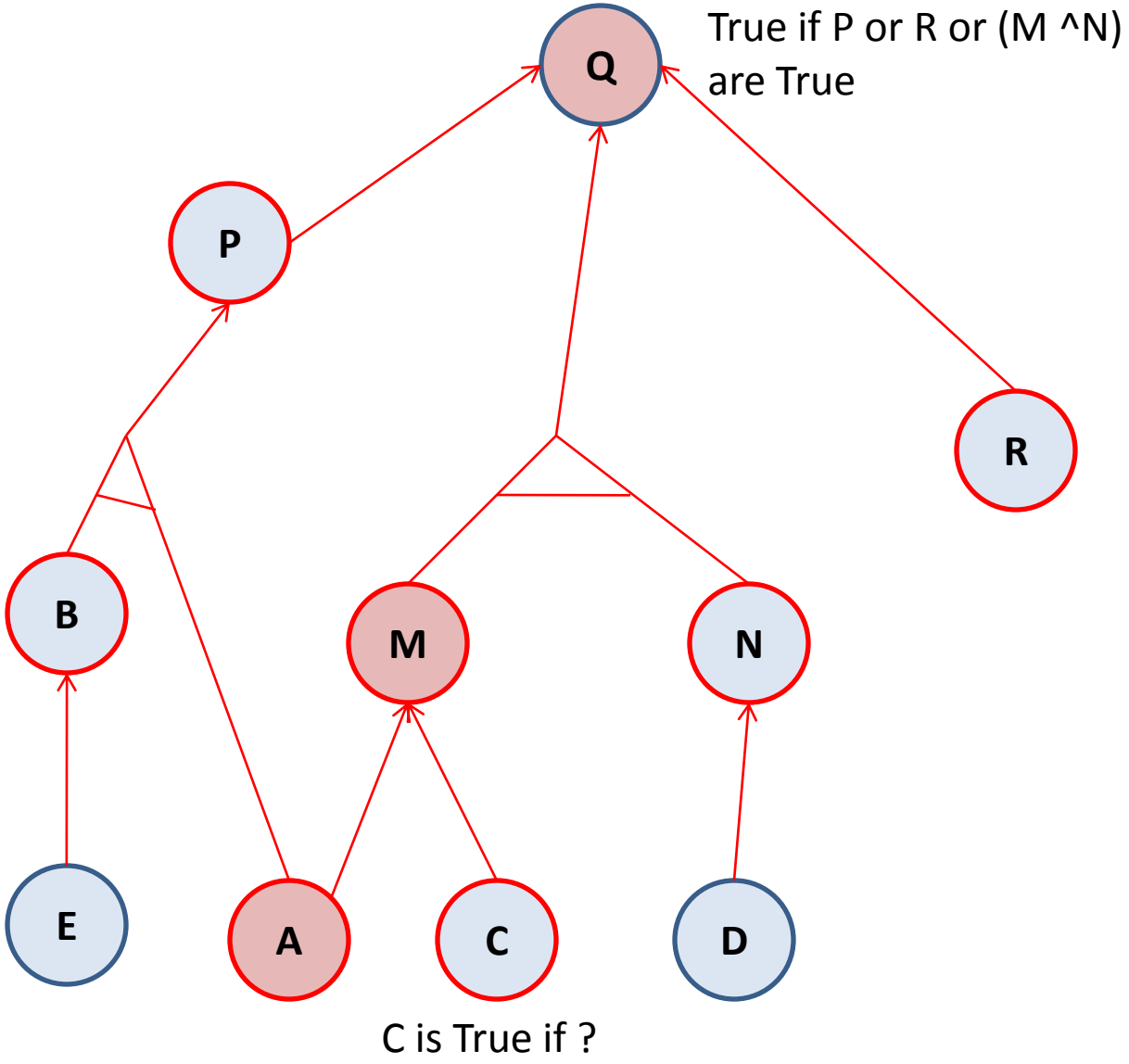
M



4. Use backward Chaining to prove Q

$P \rightarrow Q$
$E \rightarrow B$
$R \rightarrow Q$
$M \wedge N \rightarrow Q$
$A \wedge B \rightarrow P$
$A \rightarrow M$
$C \rightarrow M$
$D \rightarrow N$
D
A

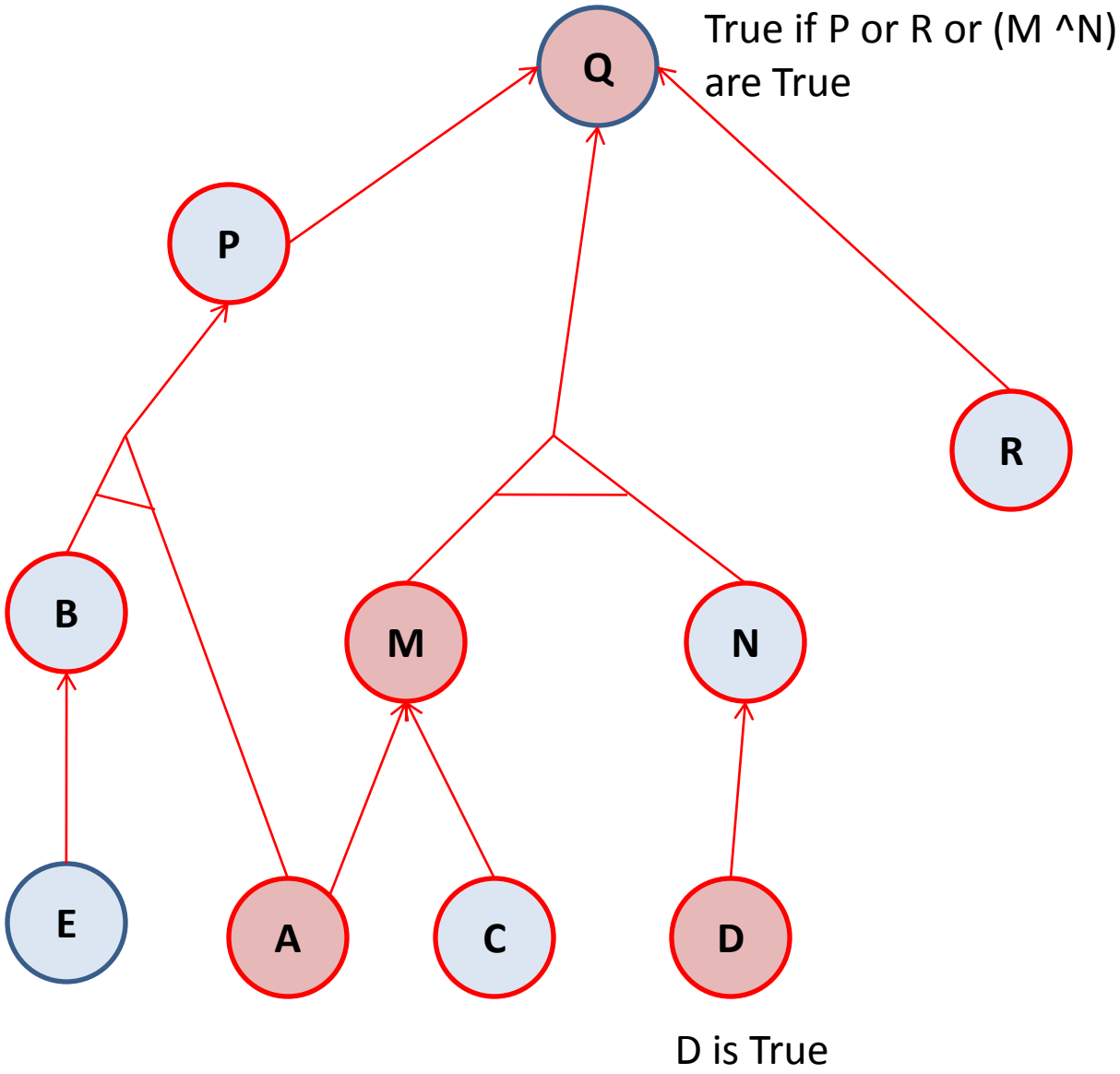
M



4. Use backward Chaining to prove Q

P → Q
E → B
R → Q
M ^ N → Q
A ^ B → P
A → M
C → M
D → N
D
A

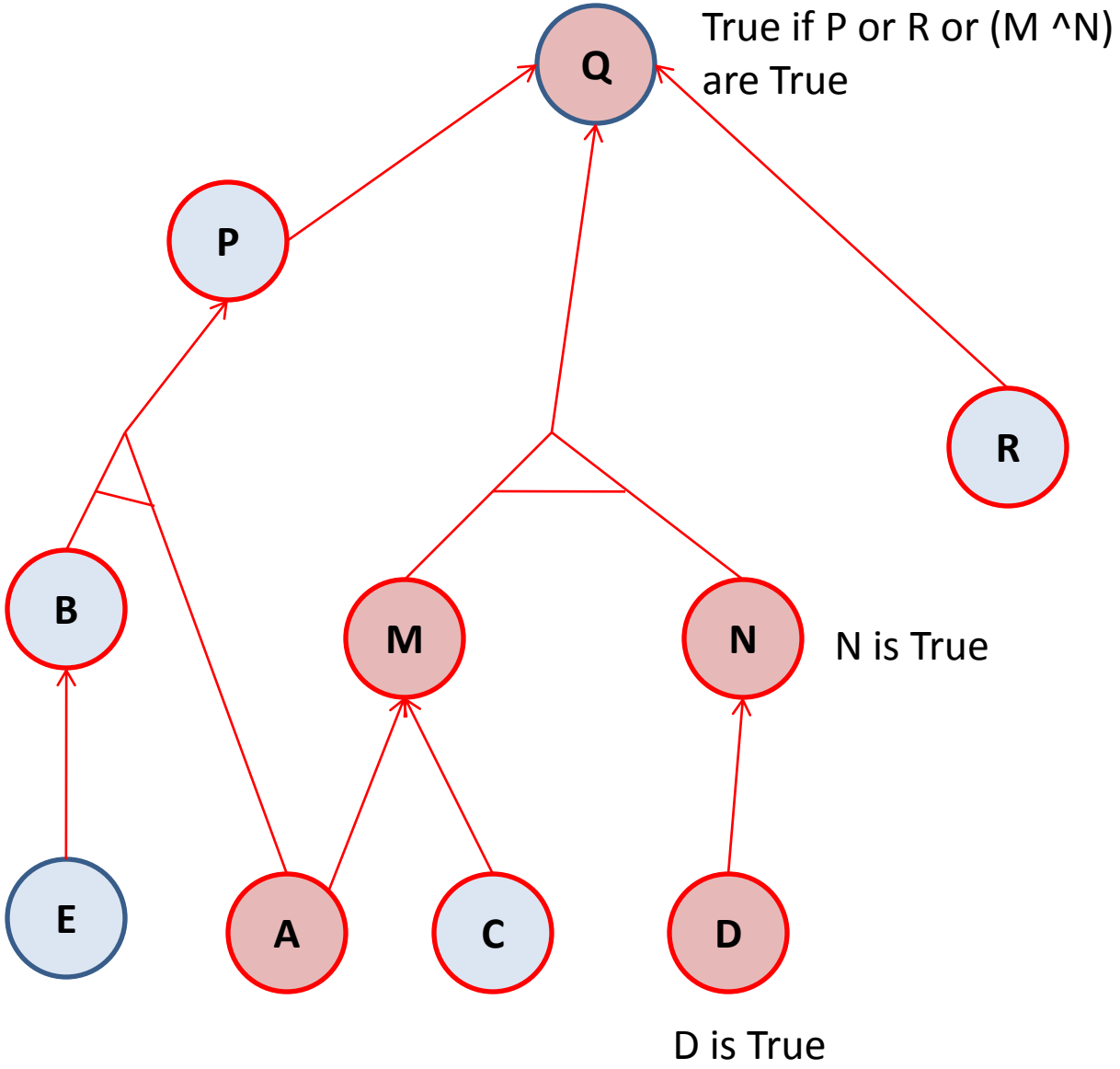
M



4. Use backward Chaining to prove Q

$P \rightarrow Q$
$E \rightarrow B$
$R \rightarrow Q$
$M \wedge N \rightarrow Q$
$A \wedge B \rightarrow P$
$A \rightarrow M$
$C \rightarrow M$
$D \rightarrow N$
D
A

M
N

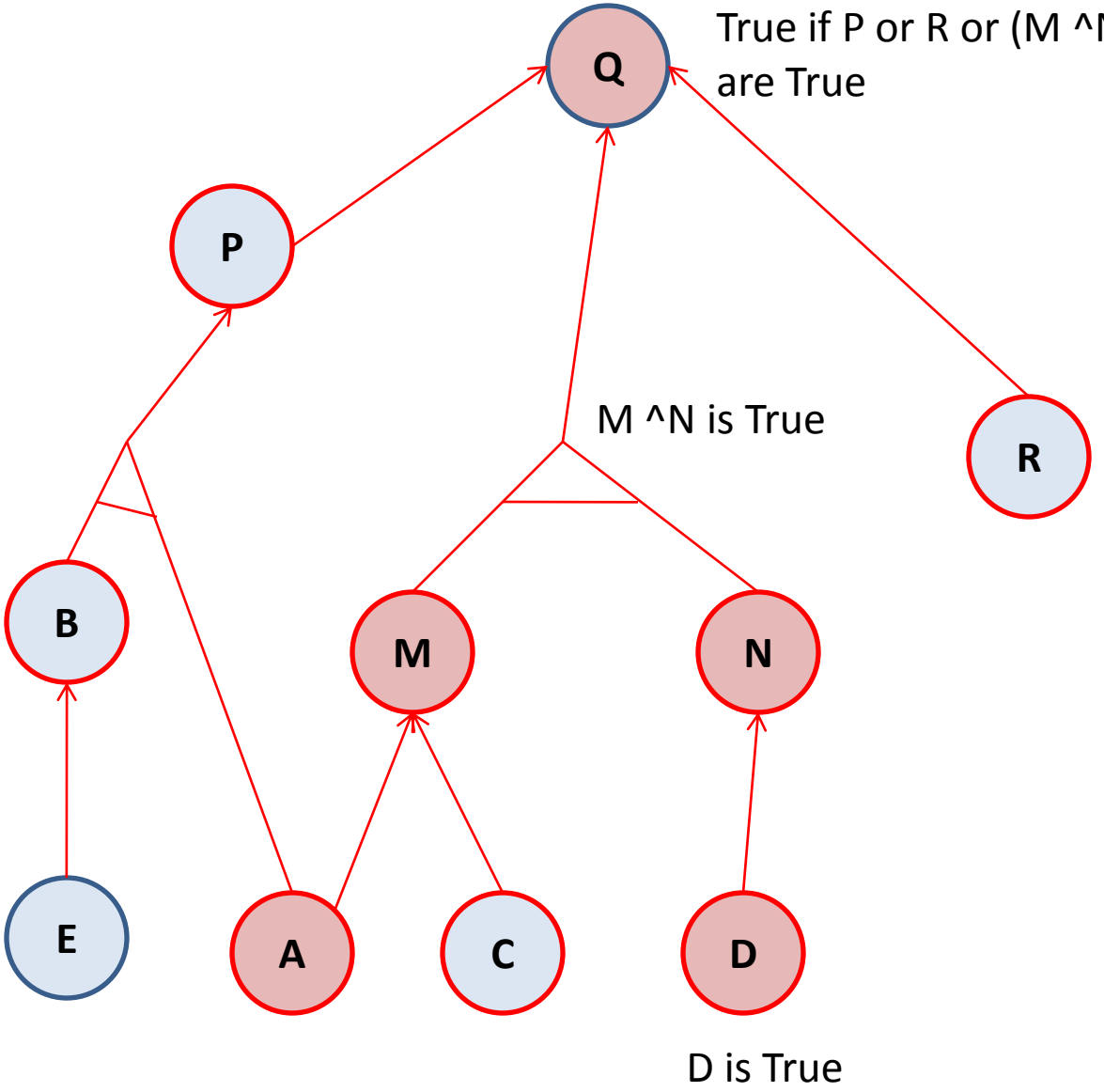


4. Use backward Chaining to prove Q

P → Q
E → B
R → Q
M ^ N → Q
A ^ B → P
A → M
C → M
D → N
D
A

M
N

M ^ N



4. Use backward Chaining to prove Q

P → Q
E → B
R → Q
M ^ N → Q
A ^ B → P
A → M
C → M
D → N
D
A

M
N

M^N

Q