CS477 Formal Software Development Methods

Elsa L Gunter
2112 SC, UIUC
egunter@illinois.edu
http://courses.engr.illinois.edu/cs477

How to build a Reduced Ordered BDD

- Let u, l, h be bdd nodes, i be a variable index
  - var(u) gives the variable at u
  - low(u) gives node pointed to by false out-edge of u
  - high(u) gives node pointed to by true out-edge of u
- Assume to tables: T and H
- T : u -> (i,l,h)
  - init T, update (T,(u,(i,l,h))),
- H : (i,l,h) -> u - Inverse of T
  - init H, member(H,(i,l,h)), lookup(H,(i,l,h)), insert (H,((i,l,h),u)

How to build a Reduced Ordered BDD

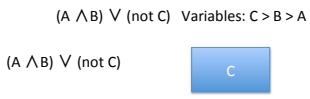
- For efficiency, use memoizing function
- function Mk[T,H](i,l,h) =

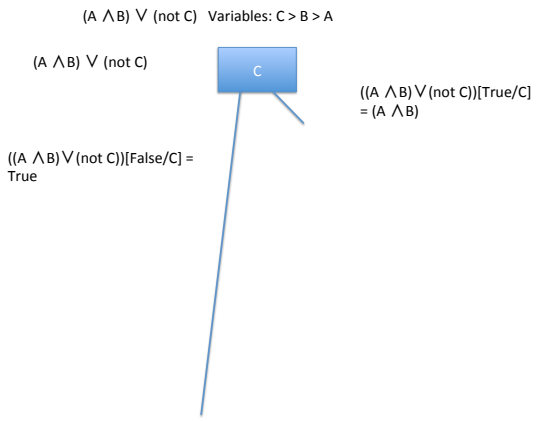
```
    if l = h then return l
    else if member (H,(i,l,h))
        then return (lookup (H,(i,l,h)))
    else update (T,(u,(i,l,h)));
        insert(H,((i,l,h),u);
        return u
end
```

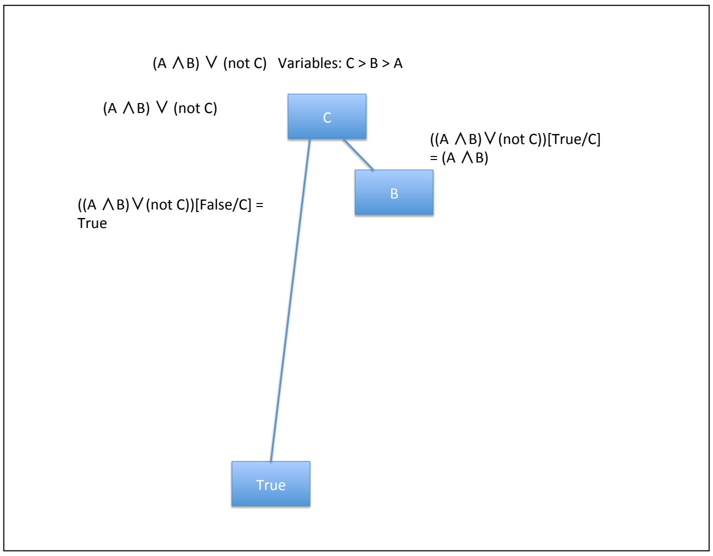
How to build a Reduced Ordered BDD

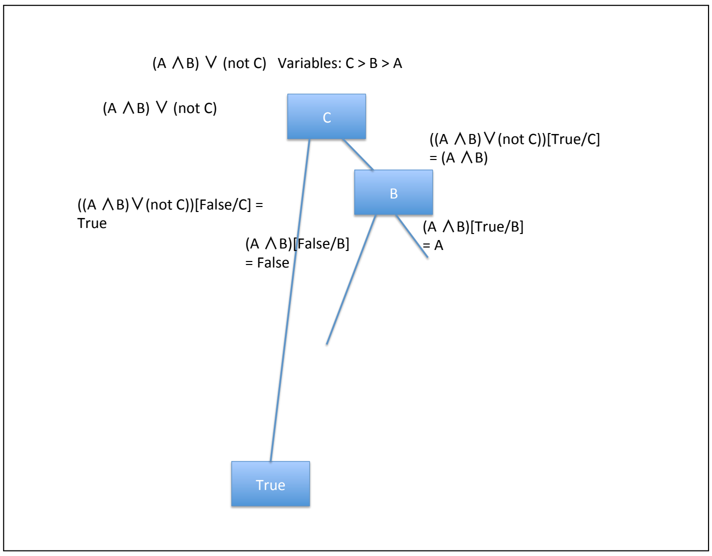
- Main function for building ROBDD
- p proposition; $x_1 > \ldots > x_n$ variables in p
- function Build[T,H](p,i) =

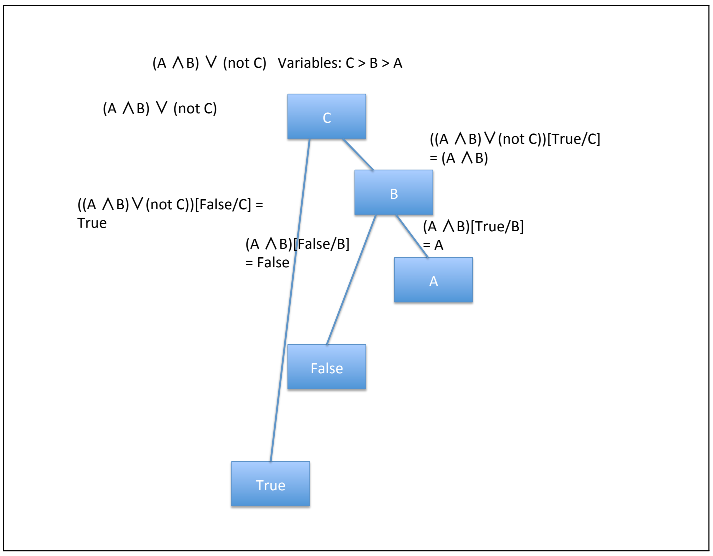
```
    if i = 0 then if p = False then False else True
    else let l = Build[T,H](p[False/x_i],(i-1)) in
        let h = Build[T,H](p[True/x_i],(oi-1)) in
        let u = Mk[T,H] (i,l,h) in
        return u
end
```

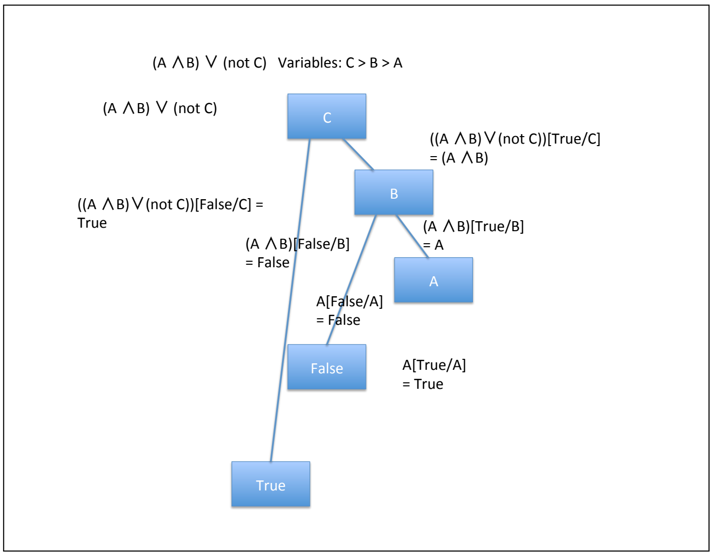


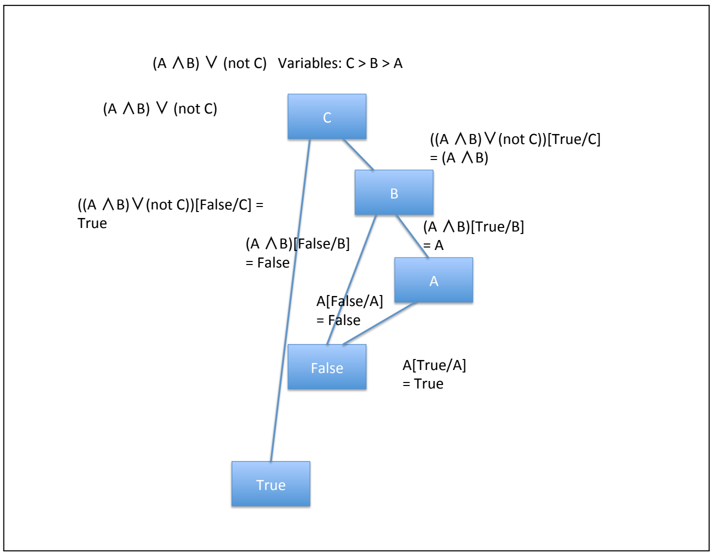


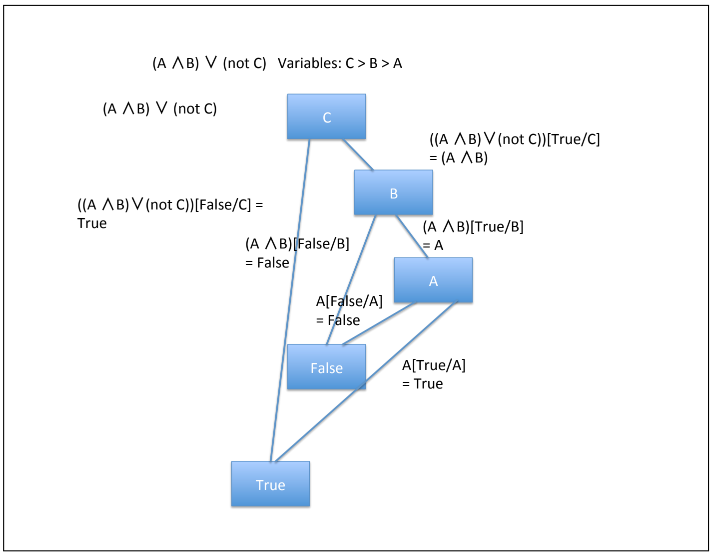

(A ∧B) ∨ (not C)   Variables: C > B > A
(A ∧B) ∨ (not C)
C
((A ∧B)∨(not C))[True/C]
= (A ∧B)
B
((A ∧B)∨(not C))[False/C] =
True
True


(A ∧B) ∨ (not C)   Variables: C > B > A
(A ∧B) ∨ (not C)
C
((A ∧B)∨(not C))[True/C]
= (A ∧B)
B
((A ∧B)∨(not C))[False/C] =
True
(A ∧B)[True/B]
= A
(A ∧B)[False/B]
= False
True


(A ∧B) ∨ (not C)   Variables: C > B > A
(A ∧B) ∨ (not C)
C
((A ∧B)∨(not C))[True/C]
= (A ∧B)
B
((A ∧B)∨(not C))[False/C] =
True
(A ∧B)[True/B]
= A
(A ∧B)[False/B]
= False
A
False
True


(A ∧B) ∨ (not C)   Variables: C > B > A
(A ∧B) ∨ (not C)
C
((A ∧B)∨(not C))[True/C]
= (A ∧B)
B
((A ∧B)∨(not C))[False/C] =
True
(A ∧B)[True/B]
= A
(A ∧B)[False/B]
= False
A
A[False/A]
= False
False
A[True/A]
= True
True


(A ∧B) ∨ (not C)   Variables: C > B > A
(A ∧B) ∨ (not C)
C
((A ∧B)∨(not C))[True/C]
= (A ∧B)
B
((A ∧B)∨(not C))[False/C] =
True
(A ∧B)[True/B]
= A
(A ∧B)[False/B]
= False
A
A[False/A]
= False
False
A[True/A]
= True
True


(A ∧B) ∨ (not C)   Variables: C > B > A
(A ∧B) ∨ (not C)
C
((A ∧B)∨(not C))[True/C]
= (A ∧B)
B
((A ∧B)∨(not C))[False/C] =
True
(A ∧B)[True/B]
= A
(A ∧B)[False/B]
= False
A
A[False/A]
= False
False
A[True/A]
= True
True

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[False/C] =
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ False))) =
(A ➔ B) ∧ ((not B) ➔ (not False)) =
(A ➔ B) ∧ ((not B) ➔ True) =
(A ➔ B) ∧ True = (A ➔ B)

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[False/C] =
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ False))) =
(A ➔ B) ∧ ((not B) ➔ (not False)) =
(A ➔ B) ∧ ((not B) ➔ True) =
(A ➔ B) ∧ True = (A ➔ B)

(A ➔ B) [False/B] =
(A ➔ False) = not A

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[False/C] =
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ False))) =
(A ➔ B) ∧ ((not B) ➔ (not False)) =
(A ➔ B) ∧ ((not B) ➔ True) =
(A ➔ B) ∧ True = (A ➔ B)

(A ➔ B) [False/B] =
(A ➔ False) = not A

(not A)[False/A] = True
(not A)[True/A] = False

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[False/C] =
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ False))) =
(A ➔ B) ∧ ((not B) ➔ (not False)) =
(A ➔ B) ∧ ((not B) ➔ True) =
(A ➔ B) ∧ True = (A ➔ B)

(A ➔ B) [False/B] =
(A ➔ False) = not A

(A ➔ B) [True/B] =
(A ➔ True) = True

(not A)[False/A] = True
(not A)[True/A] = False

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[False/C] =
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ False))) =
(A ➔ B) ∧ ((not B) ➔ (not False)) =
(A ➔ B) ∧ ((not B) ➔ True) =
(A ➔ B) ∧ True = (A ➔ B)

(A ➔ B) [False/B] =
(A ➔ False) = not A

(A ➔ B) [True/B] =
(A ➔ True) = True

(not A)[False/A] = True
(not A)[True/A] = False

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[True/C]
= (A ➔ B) ∧ ((not B) ➔ (not (A ∧True))) =
(A ➔ B) ∧ ((not B) ➔ (not A))

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[False/C] =
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ False))) =
(A ➔ B) ∧ ((not B) ➔ (not False)) =
(A ➔ B) ∧ ((not B) ➔ True) =
(A ➔ B) ∧ True = (A ➔ B)

(A ➔ B) [False/B] =
(A ➔ False) = not A

(A ➔ B) [True/B] =
(A ➔ True) = True

(not A)[False/A] = True
(not A)[True/A] = False

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[True/C]
= (A ➔ B) ∧ ((not B) ➔ (not (A ∧True))) =
(A ➔ B) ∧ ((not B) ➔ (not A))

(A ➔ B) ∧ ((not B) ➔ (not A)) [False / B]=
(A ➔ False) ∧ ((not False) ➔ (not A)) =
(not A) ∧ ((not True) ➔ (not A)) =
(not A) ∧ (False ➔ (not A)) = not A

(A → B) ∧ ((not B) → (not (A ∧ C)))

(A → B) ∧ ((not B) → (not (A ∧ C)))[False/C] =
(A → B) ∧ ((not B) → (not (A ∧ False))) =
(A → B) ∧ ((not B) → (not False)) =
(A → B) ∧ ((not B) → True) =
(A → B) ∧ True = (A → B)

(A → B) [False/B] =
(A → False) = not A

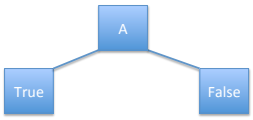
(A → B) [True/B] =
(A → True) = True

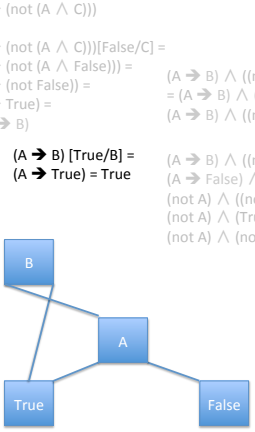
(A → B) ∧ ((not B) → (not (A ∧ C)))[True/C]
= (A → B) ∧ ((not B) → (not (A ∧True))) =
(A → B) ∧ ((not B) → (not A))

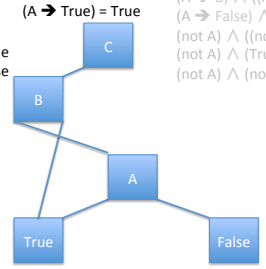
(A → B) ∧ ((not B) → (not A)) [False / B]=
(A → False) ∧ ((not False) → (not A)) =
(not A) ∧ ((not False) → (not A)) =
(not A) ∧ (True → (not A)) =
(not A) ∧ (not A)

(A → B) ∧ ((not B) →
(not A)) [True / B]=
(A → True) ∧
((not True) → (not A)) =
True ∧ (False →(not A)) =
True

(not A)[False/A] = True
(not A)[True/A] = False







(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))

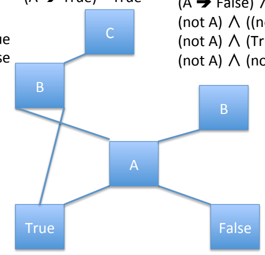
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[False/C] =
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ False))) =
(A ➔ B) ∧ ((not B) ➔ (not False)) =
(A ➔ B) ∧ ((not B) ➔ True) =
(A ➔ B) ∧ True = (A ➔ B)

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[True/C]
= (A ➔ B) ∧ ((not B) ➔ (not (A ∧True))) =
(A ➔ B) ∧ ((not B) ➔ (not A))

(A ➔ B) [False/B] =
(A ➔ False) = not A

(A ➔ B) [True/B] =
(A ➔ True) = True

(not A)[False/A] = True
(not A)[True/A] = False



(A ➔ B) ∧ ((not B) ➔ (not A)) [False / B]=
(A ➔ False) ∧ ((not False) ➔ (not A)) =
(not A) ∧ ((not False) ➔ (not A)) =
(not A) ∧ (True ➔ (not A)) =
(not A) ∧ (not A)

(A ➔ B) ∧ ((not B) ➔
(not A)) [True / B]=
(A ➔ True) ∧
((not True) ➔ (not A)) =
True ∧ (False ➔(not A)) =
True

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))

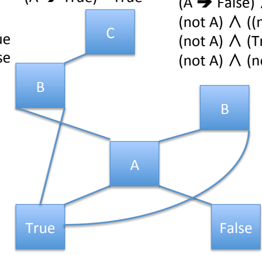
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[False/C] =
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ False))) =
(A ➔ B) ∧ ((not B) ➔ (not False)) =
(A ➔ B) ∧ ((not B) ➔ True) =
(A ➔ B) ∧ True = (A ➔ B)

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[True/C]
= (A ➔ B) ∧ ((not B) ➔ (not (A ∧True))) =
(A ➔ B) ∧ ((not B) ➔ (not A))

(A ➔ B) [False/B] =
(A ➔ False) = not A

(A ➔ B) [True/B] =
(A ➔ True) = True

(not A)[False/A] = True
(not A)[True/A] = False



(A ➔ B) ∧ ((not B) ➔ (not A)) [False / B]=
(A ➔ False) ∧ ((not False) ➔ (not A)) =
(not A) ∧ ((not False) ➔ (not A)) =
(not A) ∧ (True ➔ (not A)) =
(not A) ∧ (not A)

(A ➔ B) ∧ ((not B) ➔
(not A)) [True / B]=
(A ➔ True) ∧
((not True) ➔ (not A)) =
True ∧ (False ➔(not A)) =
True

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))

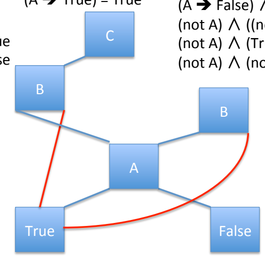
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[False/C] =
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ False))) =
(A ➔ B) ∧ ((not B) ➔ (not False)) =
(A ➔ B) ∧ ((not B) ➔ True) =
(A ➔ B) ∧ True = (A ➔ B)

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[True/C]
= (A ➔ B) ∧ ((not B) ➔ (not (A ∧True))) =
(A ➔ B) ∧ ((not B) ➔ (not A))

(A ➔ B) [False/B] =
(A ➔ False) = not A

(A ➔ B) [True/B] =
(A ➔ True) = True

(not A)[False/A] = True
(not A)[True/A] = False



(A ➔ B) ∧ ((not B) ➔ (not A)) [False / B]=
(A ➔ False) ∧ ((not False) ➔ (not A)) =
(not A) ∧ ((not False) ➔ (not A)) =
(not A) ∧ (True ➔ (not A)) =
(not A) ∧ (not A)

(A ➔ B) ∧ ((not B) ➔
(not A)) [True / B]=
(A ➔ True) ∧
((not True) ➔ (not A)) =
True ∧ (False ➔(not A)) =
True

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))

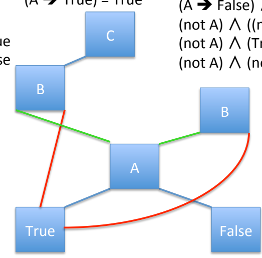
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[False/C] =
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ False))) =
(A ➔ B) ∧ ((not B) ➔ (not False)) =
(A ➔ B) ∧ ((not B) ➔ True) =
(A ➔ B) ∧ True = (A ➔ B)

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[True/C]
= (A ➔ B) ∧ ((not B) ➔ (not (A ∧True))) =
(A ➔ B) ∧ ((not B) ➔ (not A))

(A ➔ B) [False/B] =
(A ➔ False) = not A

(A ➔ B) [True/B] =
(A ➔ True) = True

(not A)[False/A] = True
(not A)[True/A] = False



(A ➔ B) ∧ ((not B) ➔ (not A)) [False / B]=
(A ➔ False) ∧ ((not False) ➔ (not A)) =
(not A) ∧ ((not False) ➔ (not A)) =
(not A) ∧ (True ➔ (not A)) =
(not A) ∧ (not A)

(A ➔ B) ∧ ((not B) ➔
(not A)) [True / B]=
(A ➔ True) ∧
((not True) ➔ (not A)) =
True ∧ (False ➔(not A)) =
True

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))

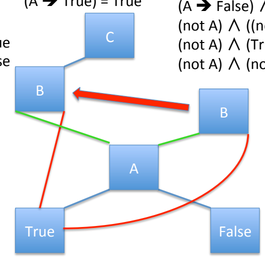
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[False/C] =
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ False))) =
(A ➔ B) ∧ ((not B) ➔ (not False)) =
(A ➔ B) ∧ ((not B) ➔ True) =
(A ➔ B) ∧ True = (A ➔ B)

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[True/C]
= (A ➔ B) ∧ ((not B) ➔ (not (A ∧True))) =
(A ➔ B) ∧ ((not B) ➔ (not A))

(A ➔ B) [False/B] =
(A ➔ False) = not A

(A ➔ B) [True/B] =
(A ➔ True) = True

(not A)[False/A] = True
(not A)[True/A] = False



(A ➔ B) ∧ ((not B) ➔ (not A)) [False / B]=
(A ➔ False) ∧ ((not False) ➔ (not A)) =
(not A) ∧ ((not False) ➔ (not A)) =
(not A) ∧ (True ➔ (not A)) =
(not A) ∧ (not A)

(A ➔ B) ∧ ((not B) ➔
(not A)) [True / B]=
(A ➔ True) ∧
((not True) ➔ (not A)) =
True ∧ (False ➔(not A)) =
True

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))

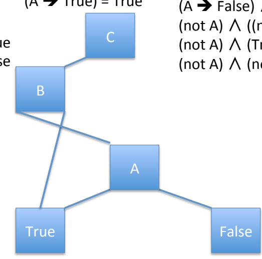
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[False/C] =
(A ➔ B) ∧ ((not B) ➔ (not (A ∧ False))) =
(A ➔ B) ∧ ((not B) ➔ (not False)) =
(A ➔ B) ∧ ((not B) ➔ True) =
(A ➔ B) ∧ True = (A ➔ B)

(A ➔ B) ∧ ((not B) ➔ (not (A ∧ C)))[True/C]
= (A ➔ B) ∧ ((not B) ➔ (not (A ∧True))) =
(A ➔ B) ∧ ((not B) ➔ (not A))

(A ➔ B) [False/B] =
(A ➔ False) = not A

(A ➔ B) [True/B] =
(A ➔ True) = True

(not A)[False/A] = True
(not A)[True/A] = False



(A ➔ B) ∧ ((not B) ➔ (not A)) [False / B]=
(A ➔ False) ∧ ((not False) ➔ (not A)) =
(not A) ∧ ((not False) ➔ (not A)) =
(not A) ∧ (True ➔ (not A)) =
(not A) ∧ (not A)

(A ➔ B) ∧ ((not B) ➔
(not A)) [True / B]=
(A ➔ True) ∧
((not True) ➔ (not A)) =
True ∧ (False ➔(not A)) =
True

(A → B) ∧ ((not B) → (not (A ∧ C)))

(A → B) ∧ ((not B) → (not (A ∧ C)))[False/C] =
(A → B) ∧ ((not B) → (not (A ∧ False))) =
(A → B) ∧ ((not B) → (not False)) =
(A → B) ∧ ((not B) → True) =
(A → B) ∧ True = (A → B)

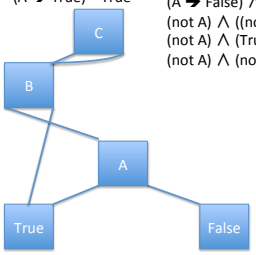
(A → B) ∧ ((not B) → (not (A ∧ C)))[True/C]
= (A → B) ∧ ((not B) → (not (A ∧True))) =
(A → B) ∧ ((not B) → (not A))

(A → B) [False/B] =
(A → False) = not A

(A → B) [True/B] =
(A → True) = True

(A → B) ∧ ((not B) → (not A)) [False / B]=
(A → False) ∧ ((not False) → (not A)) =
(not A) ∧ ((not False) → (not A)) =
(not A) ∧ (True → (not A)) =
(not A) ∧ (not A)

(not A)[False/A] = True
(not A)[True/A] = False



(A → B) ∧ ((not B) →
(not A)) [True / B]=
(A → True) ∧
((not True) → (not A)) =
True ∧ (False →(not A)) =
True

(A → B) ∧ ((not B) → (not (A ∧ C)))

(A → B) ∧ ((not B) → (not (A ∧ C)))[False/C] =
(A → B) ∧ ((not B) → (not (A ∧ False))) =
(A → B) ∧ ((not B) → (not False)) =
(A → B) ∧ ((not B) → True) =
(A → B) ∧ True = (A → B)

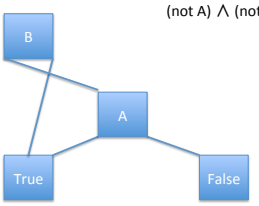
(A → B) ∧ ((not B) → (not (A ∧ C)))[True/C]
= (A → B) ∧ ((not B) → (not (A ∧True))) =
(A → B) ∧ ((not B) → (not A))

(A → B) [False/B] =
(A → False) = not A

(A → B) [True/B] =
(A → True) = True

(A → B) ∧ ((not B) → (not A)) [False / B]=
(A → False) ∧ ((not False) → (not A)) =
(not A) ∧ ((not False) → (not A)) =
(not A) ∧ (True → (not A)) =
(not A) ∧ (not A)

(not A)[False/A] = True
(not A)[True/A] = False



(A → B) ∧ ((not B) →
(not A)) [True / B]=
(A → True) ∧
((not True) → (not A)) =
True ∧ (False →(not A)) =
True