Propositional Logic II

Adapted From:

NYU G22.2390-001, Propositional Logic
Stanford CS 103, Logic
教材《数理逻辑与集合论》1.4、2.1 - 2.4

Zhaoguo Wang

Propositional Logic （命题逻辑）

1.1 Propositional Logic

Step 0. Proof.

Step 1. Convert program into mathematical formula.

Step 2. Ask the computer to solve the formula.

1.2 First Order Logic

Step 1. Convert it into first logic formula.

Step 2. Ask the computer to solve the formula.

1.3 Interactive Proof

Step 1. Axiom system

Step 2. Ask the computer to check the invariants

A Quick Recap

A propositional formula includes variables and connectives.

Propositional variables: P, Q …

Propositional connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$

WFF: construct the legal propositional formula.

Are these enough?

Can We Introduce New Connective?

We are able to define a new connective, and use truth table to describe its semantic.

P	Q	$P \triangledown Q$
T	T	F
F	T	T
T	F	T
F	F	F

Can We Introduce New Connective?

We are able to define a new connective, and use truth table to describe its semantic.

P	Q	$P \overline{\vee} Q$
T	T	F
F	T	T
T	F	T
F	F	F

$$P \overline{\vee} Q = ((P \wedge (\neg Q)) \vee ((\neg P) \wedge Q))$$

Because it can be represented with a WFF.

Completeness Of Connectives （联结词完备性）

DEFINITION

A group of connectives are **complete** if

1). Every formula can be converted to an equivalent formula and

2). The formula only includes connectives in the group.

Completeness Of Connectives（联结词完备性）

OBSERVATION

A formula with n variables can be treated to be a function:

1) The input is the truth values of n propositional variables.

2) The output is the truth value of the formula.

Completeness Of Connectives （联结词完备性）

OBSERVATION

A formula with n variables can be treated to be a function:

1) The input is the truth values of n propositional variables.

If every such function can be converted to a wff, we can prove completeness.

Completeness Of Connectives （联结词完备性）

THEOREM

Let G be an n-place Boolean function, $n \geq 1$. There exists a *WFF* α such that α realizes the function G.

DEFINITIONS

n-place Boolean function: a function with n parameters and a Boolean return value.

α realizes the function G: α and G have the same truth values.

Completeness Of Connectives （联结词完备性）

THEOREM

Let G be an n-place Boolean function, $n \geq 1$. There exists a *WFF* α such that α realizes the function G.

How to reason it out?

n-place Boolean fu… …ean return value.

α realizes the function G: α and G have the same truth value

Completeness Of Connectives
（联结词完备性）

THEOREM

Let G be an n-place Boolean function, $n \geq 1$. There exists a *WFF* α

such that α realizes the function G.

Let's start with a simple case (i.e. $n = 2$).

n-place Boolean fu ... ean

return value.

α realizes the function G: α and G have the same truth value

Completeness Of Connectives（联结词完备性）

How many connectives we can define when n is 2.

Completeness Of Connectives
（联结词完备性）

P	Q	$g_0(P,Q)$	$g_1(P,Q)$	$g_2(P,Q)$	$g_3(P,Q)$	$g_4(P,Q)$	$g_5(P,Q)$	$g_6(P,Q)$
F	F	F	F	F	F	F	F	F
F	T	F	F	F	F	T	T	T
T	F	F	F	T	T	F	F	T
T	T	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	T

$g_7(P,Q)$	$g_8(P,Q)$	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	(P,Q)
F	T	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	T
T	F	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	T
T	F	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
T	F	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

图 2.4.2

Completeness Of Connectives （联结词完备性）

P	Q	$g_0(P,Q)$	$g_1(P,Q)$	$g_2(P,Q)$	$g_3(P,Q)$	$g_4(P,Q)$	$g_5(P,Q)$	$g_6(P,Q)$
F	F	F	F	F	F	F	F	F
F	T	F	F	F	F	T	T	T
T	F	F	F	T	T	F	F	T
T	T	F	T	F	T	F	T	T

$g_7(P,Q)$	$g_8(P,Q)$	$g_9(P,Q)$	$g_{10}(P,Q)$	$g_{11}(P,Q)$	$g_{12}(P,Q)$	$g_{13}(P,Q)$	$g_{14}(P,Q)$	$g_{15}(P,Q)$
F	T	T	T	T	T	T	T	T
T	F	F	F	F	T	T	T	T
T	F	F	T	T	F	F	T	[illegible]
T	F	T	F	T	F	T	F	[illegible]

Exercise, What about g_{10} ?

Completeness Of Connectives
（联结词完备性）

P	Q	$g_0(P,Q)$	$g_1(P,Q)$
F	F	T	T
F	T	T	T
T	F	F	F
T	T	T	F

Completeness Of Connectives （联结词完备性）

P	Q	$g_0(P, Q)$	$g_1(P, Q)$
F	F	T	T
F	T	T	T
T	F	F	F
T	T	T	F

Can we derive general rules ???

Completeness Of Connectives

P	Q	$g_0(P, Q)$	$g_1(P, Q)$
F	F	$\boxed{T}$	T
F	T	$\boxed{T}$	T
T	F	F	F
T	T	$\boxed{T}$	F

Step1: find all true rows

Completeness Of Connectives

P	Q	$g_0(P, Q)$	$g_1(P, Q)$
F	F	$\boxed{T}$	T
F	T	$\boxed{T}$	T
T	F	F	F
T	T	$\boxed{T}$	F

Step1: find all true rows

Step2: generate a formula for every row

$(\neg P) \wedge (\neg Q)$ $\quad$ $(\neg P) \wedge Q$ $\quad$ $P \wedge Q$

Completeness Of Connectives

P	Q	$g_0(P, Q)$	$g_1(P, Q)$
F	F	T	T
F	T	T	T
T	F	F	F
T	T	T	F

Step1: find all true rows

Step2: generate a formula for every row

Step3: use "or" to connect these formulas

$$g_0(P, Q) = (((\neg P) \wedge (\neg Q)) \vee ((\neg P) \wedge Q)) \vee (P \wedge Q)$$

Completeness Of Connectives

P	Q	$g_0(P,Q)$	$g_1(P,Q)$
F	F	T	T
F	T	T	T
T	F	F	F
T	T	T	F

Step1: find all true rows

Step2: generate a formula for every row

Step3: use "or" to connect these formulas

What about g1(P,Q)?

$$g_0(P,Q) = (((\neg P) \wedge (\neg Q)) \vee ((\neg P) \wedge Q)) \vee (P \wedge Q)$$

$$g_1(P,Q) = ((\neg P) \wedge (\neg Q)) \vee ((\neg P) \wedge Q)$$

Completeness Of Connectives

P	Q	$g_0(P,Q)$	$g_1(P,Q)$
F	F	T	T
F	T	T	T
T	F	F	F
T	T	T	F

Another algorithm

Completeness Of Connectives

P	Q	$g_0(P, Q)$	$g_1(P, Q)$
F	F	T	T
F	T	T	T
T	F	F	$\boxed{F}$
T	T	T	$\boxed{F}$

Step1: find all false rows

Completeness Of Connectives

P	Q	$g_0(P,Q)$	$g_1(P,Q)$
F	F	T	T
F	T	T	T
T	F	F	$\boxed{\text{F}}$
T	T	T	$\boxed{\text{F}}$

Step1: find all false rows

Step2: generate a formula for every row

$((\neg P) \vee Q) \quad ((\neg P) \vee (\neg Q))$

Completeness Of Connectives

P	Q	$g_0(P,Q)$	$g_1(P,Q)$
F	F	T	T
F	T	T	T
T	F	F	F
T	T	T	F

Step1: find all false rows

Step2: generate a formula for every row

Step3: use "and" to connect these formulas

$$g_1(P,Q) = ((\neg P) \vee Q) \wedge ((\neg P) \vee (\neg Q))$$

Completeness Of Connectives

P	Q	$g_0(P, Q)$	$g_1(P, Q)$
F	F	T	T
F	T	T	T
T	F	$\boxed{\text{F}}$	F
T	T	T	F

Step1: find all false rows

Step2: generate a formula for every row

Step3: use "and" to connect these formulas

What about g0(P,Q)?

$$g_1(P, Q) = ((\neg P) \vee Q) \wedge ((\neg P) \vee (\neg Q))$$

$$g_0(P, Q) = ((\neg P) \vee Q)$$

Completeness Of Connectives

P	Q	$g_0(P, Q)$	$g_1(P, Q)$
F	F	$\boxed{\text{T}}$	$\boxed{\text{T}}$
F	T	$\boxed{\text{T}}$	$\boxed{\text{T}}$
T	F	F	F
T	T	$\boxed{\text{T}}$	F

P	Q	$g_0(P, Q)$	$g_1(P, Q)$
F	F	T	T
F	T	T	T
T	F	F	F
T	T	T	F

$$g_0(P, Q) = (((\neg P) \wedge (\neg Q)) \vee ((\neg P) \wedge Q)) \vee (P \wedge Q)$$

$$g_1(P, Q) = ((\neg P) \wedge (\neg Q)) \vee ((\neg P) \wedge Q)$$

Completeness Of Connectives

P	Q	$g_0(P,Q)$	$g_1(P,Q)$
F	F	$\boxed{T}$	$\boxed{T}$
F	T	$\boxed{T}$	$\boxed{T}$
T	F	F	F
T	T	$\boxed{T}$	F

P	Q	$g_0(P,Q)$	$g_1(P,Q)$
F	F	T	T
F	T	T	T
T	F	$\boxed{F}$	$\boxed{F}$
T	T	T	$\boxed{F}$

$$g_0(P,Q) = (((\neg P) \wedge (\neg Q)) \vee ((\neg P) \wedge Q)) \vee (P \wedge Q)$$

$$g_1(P,Q) = ((\neg P) \wedge (\neg Q)) \vee ((\neg P) \wedge Q)$$

$$g_0(P,Q) = ((\neg P) \vee Q)$$

$$g_1(P,Q) = ((\neg P) \vee Q) \wedge ((\neg P) \vee (\neg Q))$$

Completeness Of Connectives

P	Q	$g_0(P,Q)$	$g_1(P,Q)$
F	F	$\boxed{T}$	$\boxed{T}$
F	T	$\boxed{T}$	$\boxed{T}$
T	F	F	F
T	T	$\boxed{T}$	F

P	Q	$g_0(P,Q)$	$g_1(P,Q)$
F	F	T	T
F	T	T	T
T	F	$\boxed{F}$	$\boxed{F}$
T	T	T	$\boxed{F}$

$$g_0(P,Q) = (((\neg P) \wedge (\neg Q)) \vee ((\neg P) \wedge Q)) \vee (P \wedge Q)$$

$$g_1(P,Q) = ((\neg P) \wedge (\neg Q)) \vee ((\neg P) \wedge Q)$$

$$g_0(P,Q) = ((\neg P) \vee Q)$$

$$g_1(P,Q) = ((\neg P) \vee Q) \wedge ((\neg P) \vee (\neg Q))$$

M1. According to the T rows

M2. According to the F rows

Completeness Of Connectives
（联结词完备性）

THEOREM

Let G be an n-place Boolean function, $n \geq 1$. There exists a *WFF* α such that α realizes the function G.

DEFINITIONS

n- … arameters and a Boolean

ret…

α re… ame t… lues.

We now give a formal proof

Completeness Of Connectives （联结词完备性）

THEOREM

Let G be an n-place Boolean function, $n \geq 1$. There exists a *WFF* α such that α realizes the function G.

Proof

If G always return F, then $\alpha = P \wedge (\neg P)$. It is clear α realizes the function G.

Otherwise, G returns true sometimes. Suppose there are k cases where G returns true.

Completeness Of Connectives

Proof

There are k inputs that can make G return true:

G(X_{11}, X_{12}, …, X_{1n}) = T

G(X_{21}, X_{22}, …, X_{2n}) = T

…

G(X_{k1}, X_{k2}, …, X_{kn}) = T

Then we can construct α in the following way.

$$\beta_{ij} = \begin{cases} P_j & if \;\; X_{ij} = T \\ \neg P_j & if \;\; X_{ij} = F \end{cases}$$

$$\gamma_i = \beta_{i1} \wedge \cdots \wedge \beta_{in}$$

$$\alpha = \gamma_1 \vee \cdots \vee \gamma_k$$

Looks like algorithm above

Completeness Of Connectives

Proof

When G returns true, it is clear that α is true.
When α is true, there must be some γ_i is true.
There is only one assignment that can make γ_i is true.
Under this assignment, G returns true.

G($X_{11}, X_{12}, \ldots, X_{1n}$) = T
G($X_{21}, X_{22}, \ldots, X_{2n}$) = T
…
G($X_{k1}, X_{k2}, \ldots, X_{kn}$) = T

$$\beta_{ij} = \begin{cases} P_j & if\ \ X_{ij} = T \\ \neg P_j & if\ \ X_{ij} = F \end{cases}$$

$$\gamma_i = \beta_{i1} \wedge \cdots \wedge \beta_{in}$$

$$\alpha = \gamma_1 \vee \cdots \vee \gamma_k$$

A Quick Recap

A propositional formula includes variables and connectives.

Propositional variables: P, Q …

Propositional connectives: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$

WFF: construct the legal propositional formula.

Are they optimal???

Completeness Of Connectives

We already know that

$$P \leftrightarrow Q = (P \rightarrow Q) \wedge (Q \rightarrow P)$$

The group becomes {¬, ∧, ∨, →}

What about "→" ?

Completeness Of Connectives

P	Q	$P \to Q$	$(\neg P) \vee Q$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

Implication can be expressed by "not" and "or"

The group becomes $\{\neg, \wedge, \vee\}$

Completeness Of Connectives

P	Q	$P \wedge Q$	$\neg((\neg P) \vee (\neg Q))$
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	F

"and" can be expressed by "not" and "or"

P	Q	$P \vee Q$	$\neg((\neg P) \wedge (\neg Q))$
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	F

"or" can be expressed by "not" and "and"

Finally, $\{\neg, \wedge\}$ and $\{\neg, \vee\}$ are all complete

We cannot eliminate elements from them.

Now, can you use the propositional logic to formalize the "2 queens problem theorem?"

Example

P1	P2
P3	P4

Theorem: Two queens problem has no solution.

REPHRASE

2 queens problem: placing 2 chess queens on an 2×2 chessboard so that no two queens share the same row, column or diagonal.

How to formalize the problem has no solution?

Example

P1	P2
P3	P4

Theorem: Two queens problem has no solution.

REPHRASE

2 queens problem: placing 2 chess queens on an 2×2 chessboard so that no two queens share the same row, column or diagonal.

There are four positions on the chessboard. We now define four propositional variables P1, P2, P3 and P4.

$P1$ signifies that there is a queen on position 1. $\neg P1$ signifies there is not a queen on position 1 and so on.

Example

Theorem: Two queens problem has no solution.

P1	P2
P3	P4

REPHRASE

2 queens problem: placing 2 chess queens on an 2×2 chessboard so that no two queens share the same row, column or diagonal.

First, we formalize the requirement:

$$(((((\neg(P1 \wedge P2)) \wedge (\neg(P1 \wedge P3))) \wedge$$
$$(\neg(P1 \wedge P4))) \wedge (\neg(P2 \wedge P3))) \wedge$$
$$(\neg(P2 \wedge P4))) \wedge (\neg(P3 \wedge P4)))$$

Example

P1	P2
P3	P4

Theorem: Two queens problem has no solution.

REPHRASE

2 queens problem: placing 2 chess queens on an 2×2 chessboard so that no two queens share the same row, column or diagonal.

Second, we formalize the placement:

$$\Big(\big((P1 \wedge P2) \wedge (\neg P3)\big) \wedge (\neg P4)\Big) \vee \Big(\big((P1 \wedge P3) \wedge (\neg P2)\big) \wedge (\neg P4)\Big) \vee$$
$$\Big(\big((P1 \wedge P4) \wedge (\neg P2)\big) \wedge (\neg P3)\Big) \vee \Big(\big((P2 \wedge P3) \wedge (\neg P1)\big) \wedge (\neg P4)\Big) \vee$$
$$\Big(\big((P2 \wedge P4) \wedge (\neg P3)\big) \wedge (\neg P1)\Big) \vee \Big(\big((P3 \wedge P4) \wedge (\neg P1)\big) \wedge (\neg P2)\Big)$$

Example

P1	P2
P3	P4

Theorem: Two queens problem has no solution.

REPHRASE

2 queens problem: placing 2 chess queens on an 2×2 chessboard so that no two queens share the same row, column or diagonal.

$$\left[\begin{array}{l} (((((\neg(P1 \wedge P2)) \wedge (\neg(P1 \wedge P3)))\wedge \\ (\neg(P1 \wedge P4))) \wedge (\neg(P2 \wedge P3)))\wedge \\ (\neg(P2 \wedge P4))) \wedge (\neg(P3 \wedge P4))) \end{array} \wedge \begin{array}{l} (((P1 \wedge P2) \wedge (\neg P3)) \wedge (\neg P4))\vee \\ (((P1 \wedge (\neg P2)) \wedge P3) \wedge (\neg P4))\vee \\ (((P1 \wedge (\neg P2)) \wedge (\neg P3)) \wedge P4)\vee \\ ((((\neg P1) \wedge P2) \wedge P3) \wedge (\neg P4))\vee \\ ((((\neg P1) \wedge P2) \wedge (\neg P3)) \wedge P4)\vee \\ ((((\neg P1) \wedge (\neg P2)) \wedge P3) \wedge P4) \end{array}\right] \leftrightarrow F$$

$$\left[\begin{array}{l}(((((\neg(P1 \wedge P2)) \wedge (\neg(P1 \wedge P3)))\wedge \\ (\neg(P1 \wedge P4))) \wedge (\neg(P2 \wedge P3)))\wedge \\ (\neg(P2 \wedge P4))) \wedge (\neg(P3 \wedge P4)))\end{array} \quad \wedge \quad \begin{array}{l}(((P1 \wedge P2) \wedge (\neg P3)) \wedge (\neg P4))\vee \\ (((P1 \wedge (\neg P2)) \wedge P3) \wedge (\neg P4))\vee \\ (((P1 \wedge (\neg P2)) \wedge (\neg P3)) \wedge P4)\vee \\ ((((\neg P1) \wedge P2) \wedge P3) \wedge (\neg P4))\vee \\ ((((\neg P1) \wedge P2) \wedge (\neg P3)) \wedge P4)\vee \\ ((((\neg P1) \wedge (\neg P2)) \wedge P3) \wedge P4)\end{array}\right] \leftrightarrow F$$

How to prove it?

$$
\left(
\begin{array}{l}
(((((\neg(P1 \wedge P2)) \wedge (\neg(P1 \wedge P3)))\wedge \\
(\neg(P1 \wedge P4))) \wedge (\neg(P2 \wedge P3)))\wedge \\
(\neg(P2 \wedge P4))) \wedge (\neg(P3 \wedge P4)))
\end{array}
\wedge
\begin{array}{l}
(((P1 \wedge P2) \wedge (\neg P3)) \wedge (\neg P4))\vee \\
(((P1 \wedge (\neg P2)) \wedge P3) \wedge (\neg P4))\vee \\
(((P1 \wedge (\neg P2)) \wedge (\neg P3)) \wedge P4)\vee \\
((((\neg P1) \wedge P2) \wedge P3) \wedge (\neg P4))\vee \\
((((\neg P1) \wedge P2) \wedge (\neg P3)) \wedge P4)\vee \\
((((\neg P1) \wedge (\neg P2)) \wedge P3) \wedge P4)
\end{array}
\right) \leftrightarrow F
$$

How to prove it?
Truth table.

Truth Table

There are 4 propositional variables.

P_1	...	P_4	$g(P_1, P_2, P_3, P_4)$
T	...	T	T
...	...	...	...
...	...	...	...
F	...	F	T

??? ROWS

Truth Table

There are 4 propositional variables.

P_1	...	P_4	$g(P_1, P_2, P_3, P_4)$
T	...	T	T
...	...	...	...
...	...	...	...
F	...	F	T

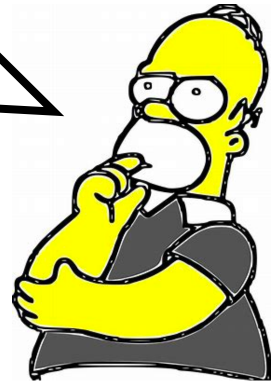
16 ROWS

If there are n variables, how many rows in the truth table?

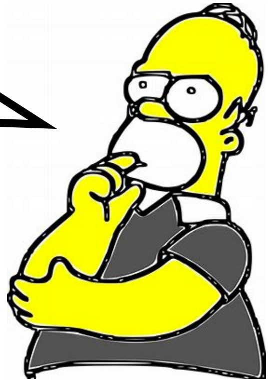
$\mathbf{2^n}$!!!

What will affect the computation cost?

The computation cost depends on #variables and the complexity of the formula.



Can we simplify the formula???



$$\left[\begin{array}{l} (((((\neg(P1 \wedge P2)) \wedge (\neg(P1 \wedge P3))) \wedge \\ (\neg(P1 \wedge P4))) \wedge (\neg(P2 \wedge P3))) \wedge \\ (\neg(P2 \wedge P4))) \wedge (\neg(P3 \wedge P4))) \end{array} \wedge \begin{array}{l} (((P1 \wedge P2) \wedge (\neg P3)) \wedge (\neg P4)) \vee \\ (((P1 \wedge (\neg P2)) \wedge P3) \wedge (\neg P4)) \vee \\ (((P1 \wedge (\neg P2)) \wedge (\neg P3)) \wedge P4) \vee \\ ((((\neg P1) \wedge P2) \wedge P3) \wedge (\neg P4)) \vee \\ ((((\neg P1) \wedge P2) \wedge (\neg P3)) \wedge P4) \vee \\ ((((\neg P1) \wedge (\neg P2)) \wedge P3) \wedge P4) \end{array} \right] \leftrightarrow F$$

Can we eliminate them firstly without affecting the semantic?

Operator Precedence

$$((P \vee Q) \wedge R)$$

$$P \vee Q \wedge R$$

$$(P \vee (Q \wedge R))$$

$P = T, Q = F, R = F$

Operator Precedence

Recap: what does $\neg(P1 \land P2)$ mean in two queens problem?

But, what does $\neg P1 \land P2$ mean?

Operator Precedence

We introduce operator precedence like that in C, C++ and python

$$\neg \;>\; \wedge \;>\; \vee \;>\; \rightarrow \;>\; \leftrightarrow$$

$\neg P1 \wedge P2$

$P \vee Q \wedge R$

Operator Precedence

Only operator precedence is not enough

$P \to Q \to R$

 $\to$  $\to$ 

Operator Precedence

The real meaning is

$$P \rightarrow (Q \rightarrow R)$$

$\rightarrow$ (

$\rightarrow$

)

Operator Precedence

But there is can be a different interpretation

$$(P \to Q) \to R$$

$\rightarrow$

$\rightarrow$

Operator Precedence

Left-associative（左结合）

✓ The same precedence are evaluated in order from left to right.

$$P \to Q \to R$$

( → ) → 

Operator Precedence

With operator precedence and association, we cannot write the following formula with less parentheses.

 $\rightarrow$ (

$\rightarrow$ )

Exercise

How to parse this statement?

$$\neg X \to Y \vee Z \to X \vee Y \wedge Z$$

Operator precedence

$$\neg \;>\; \wedge \;>\; \vee \;>\; \to \;>\; \leftrightarrow$$

All operators are left-associative

Exercise

How to parse this statement?

$$(\neg X) \to Y \lor Z \to X \lor Y \land Z$$

Operator precedence

$$\neg \;>\; \land \;>\; \lor \;>\; \to \;>\; \leftrightarrow$$

All operators are left-associative

Exercise

How to parse this statement?

$$(\neg X) \to Y \vee Z \to X \vee (Y \wedge Z)$$

Operator precedence

$$\neg \;>\; \wedge \;>\; \vee \;>\; \to \;>\; \leftrightarrow$$

All operators are left-associative

Exercise

How to parse this statement?

$$(\neg X) \rightarrow (Y \vee Z) \rightarrow (X \vee (Y \wedge Z))$$

Operator precedence

$$\neg \quad > \quad \wedge \quad > \quad \vee \quad > \quad \rightarrow \quad > \quad \leftrightarrow$$

All operators are left-associative

Exercise

How to parse this statement?

$$((\neg X) \to (Y \vee Z)) \to (X \vee (Y \wedge Z))$$

Operator precedence

$$\neg \;>\; \wedge \;>\; \vee \;>\; \to \;>\; \leftrightarrow$$

All operators are left-associative

$$\left[\begin{array}{l}(((((\neg(P1 \wedge P2)) \wedge (\neg(P1 \wedge P3)))\wedge \\ (\neg(P1 \wedge P4))) \wedge (\neg(P2 \wedge P3)))\wedge \\ (\neg(P2 \wedge P4))) \wedge (\neg(P3 \wedge P4)))\end{array} \quad \wedge \quad \begin{array}{l}(((P1 \wedge P2) \wedge (\neg P3)) \wedge (\neg P4))\vee \\ (((P1 \wedge (\neg P2)) \wedge P3) \wedge (\neg P4))\vee \\ (((P1 \wedge (\neg P2)) \wedge (\neg P3)) \wedge P4)\vee \\ ((((\neg P1) \wedge P2) \wedge P3) \wedge (\neg P4))\vee \\ ((((\neg P1) \wedge P2) \wedge (\neg P3)) \wedge P4)\vee \\ ((((\neg P1) \wedge (\neg P2)) \wedge P3) \wedge P4)\end{array}\right] \leftrightarrow F$$

Which parentheses can be eliminated?

$$\left(\begin{array}{c} (((((\neg(P1 \wedge P2)) \wedge (\neg(P1 \wedge P3)))\wedge \\ (\neg(P1 \wedge P4))) \wedge (\neg(P2 \wedge P3)))\wedge \\ (\neg(P2 \wedge P4))) \wedge (\neg(P3 \wedge P4))) \end{array} \wedge \begin{array}{c} (((P1 \wedge P2) \wedge (\neg P3)) \wedge (\neg P4))\vee \\ (((P1 \wedge (\neg P2)) \wedge P3) \wedge (\neg P4))\vee \\ (((P1 \wedge (\neg P2)) \wedge (\neg P3)) \wedge P4)\vee \\ ((((\neg P1) \wedge P2) \wedge P3) \wedge (\neg P4))\vee \\ ((((\neg P1) \wedge P2) \wedge (\neg P3)) \wedge P4)\vee \\ ((((\neg P1) \wedge (\neg P2)) \wedge P3) \wedge P4) \end{array}\right) \leftrightarrow F$$

$$(((P1 \wedge P2) \wedge (\neg P3)) \wedge (\neg P4)) \Rightarrow P1 \wedge P2 \wedge \neg P3 \wedge \neg P4$$

What about this? $\neg(P1 \wedge P2)$

Propositional Equivalences（等值）

Theorem

Propositional Equivalences（等值）： If two formula P and Q has the same truth value under any assignment, they are equivalent. It is denoted by $P = Q$ or $P \Leftrightarrow Q$.

Equivalence Theorem（等值定理）： $P = Q$ iff $P \leftrightarrow Q$ is always true.

De Morgan‘s Laws（摩根律）

Theorem

De Morgan’s Laws:

$$\neg(P \wedge Q) = \neg P \vee \neg Q, \neg(P \vee Q) = \neg P \wedge \neg Q$$

$\neg(P1 \wedge P2)$ can be replaced by $\neg P1 \vee \neg P2$.

What does $\neg(P1 \wedge P2)$ and $\neg P1 \vee \neg P2$ mean?

More Laws

Theorem

Double Negation（双重否定律）：

$$\neg\neg P = P$$

P	Q	$P \to Q$
T	T	T
T	F	$\boxed{\text{F}}$
F	T	T
F	F	T

According to F rows, $P \longrightarrow Q = \neg P \vee Q$
and $\neg P \vee Q = \neg(P \wedge \neg Q)$.
So, $P \longrightarrow Q = \neg(P \wedge \neg Q)$

More Laws

Theorem

Associative Law（结合律）：

$$(P \vee Q) \vee R = P \vee (Q \vee R)$$
$$(P \wedge Q) \wedge R = P \wedge (Q \wedge R)$$
$$(P \leftrightarrow Q) \leftrightarrow R = P \leftrightarrow (Q \leftrightarrow R)$$

$$(P \rightarrow Q) \rightarrow R \neq P \rightarrow (Q \rightarrow R)$$

Warning!

More Laws

Theorem

Commutative Law（交换律）：

$$P \vee Q = Q \vee P$$
$$P \wedge Q = Q \wedge P$$
$$P \leftrightarrow Q = Q \leftrightarrow P$$

$P \to Q = Q \to P$?

It is wrong. Can you give a counterexample?

More Laws

Theorem

Idempotent Law（等幂率）:

$$P \vee P = P$$
$$P \wedge P = P$$
$$P \leftrightarrow P = T$$
$$P \rightarrow P = T$$

Theorem

Identity Law（同一律）:

$$P \vee F = P$$
$$P \wedge T = P$$
$$T \rightarrow P = P$$
$$T \leftrightarrow P = P$$
$$P \rightarrow F = \neg P$$
$$F \leftrightarrow P = \neg P$$

More Laws

Theorem

Complementary Law（补余律）：

$$P \vee \neg P = T$$
$$P \wedge \neg P = F$$
$$P \rightarrow \neg P = \neg P$$
$$\neg P \rightarrow P = P$$
$$P \leftrightarrow \neg P = F$$

Zero Law （零律）：

$$P \vee T = T$$
$$P \wedge F = F$$
$$P \rightarrow T = T$$
$$F \rightarrow P = T$$

More Laws

Theorem

Distributive Law（分配律）：

$$P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$$
$$P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$$

Absorption Law（吸收律）：

$$P \vee (P \wedge Q) = P$$
$$P \wedge (P \vee Q) = P$$

$$P \wedge (Q_1 \vee Q_2 \vee \ldots \vee Q_n) = ?$$

More Laws

Theorem

Distributive Law（分配律）：

$$P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$$
$$P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$$

Absorption Law（吸收律）：

$$P \vee (P \wedge Q) = P$$
$$P \wedge (P \vee Q) = P$$

$$P \wedge (Q_1 \vee Q_2 \vee \ldots \vee Q_n) = (P \wedge Q_1) \vee (P \wedge Q_2) \vee \ldots \vee (P \wedge Q_n)$$

Example

$$
\begin{array}{l}
(\\
\quad (((P1 \wedge P2) \wedge (\neg P3)) \wedge (\neg P4)) \vee \\
\quad (((P1 \wedge (\neg P2)) \wedge P3) \wedge (\neg P4)) \vee \\
\quad (((P1 \wedge (\neg P2)) \wedge (\neg P3)) \wedge P4) \vee \\
\quad ((((\neg P1) \wedge P2) \wedge P3) \wedge (\neg P4)) \vee \\
\quad ((((\neg P1) \wedge P2) \wedge (\neg P3)) \wedge P4) \vee \\
\quad ((((\neg P1) \wedge (\neg P2)) \wedge P3) \wedge P4) \\
) \wedge \\
(\ (((((\neg(P1 \wedge P2)) \wedge (\neg(P1 \wedge P3))) \wedge \\
\quad (\neg(P1 \wedge P4))) \wedge (\neg(P2 \wedge P3))) \wedge \\
\quad (\neg(P2 \wedge P4))) \wedge (\neg(P3 \wedge P4))) \\
\ \leftrightarrow F
\end{array}
$$

Eliminate redundant parentheses

Example

$$
\begin{aligned}
&(\\
&\quad (P1 \land P2 \land \neg P3 \land \neg P4)\lor\\
&\quad (P1 \land \neg P2 \land P3 \land \neg P4)\lor\\
&\quad (P1 \land \neg P2 \land \neg P3 \land P4)\lor\\
&\quad (\neg P1 \land P2 \land P3 \land \neg P4)\lor\\
&\quad (\neg P1 \land P2 \land \neg P3 \land P4)\lor\\
&\quad (\neg P1 \land \neg P2 \land P3 \land P4)\\
&)\land\\
&\quad \neg(P1 \land P2) \land \neg(P1 \land P3)\land\\
&\quad \neg(P1 \land P4) \land \neg(P2 \land P3)\land\\
&\quad \neg(P2 \land P4) \land \neg(P3 \land P4)\\
&\quad \leftrightarrow F
\end{aligned}
$$

We can Eliminate more Parentheses. But we retain them for readability.

Example

$$
\begin{aligned}
&(\\
&\quad (P1 \wedge P2 \wedge \neg P3 \wedge \neg P4) \vee\\
&\quad (P1 \wedge \neg P2 \wedge P3 \wedge \neg P4) \vee\\
&\quad (P1 \wedge \neg P2 \wedge \neg P3 \wedge P4) \vee\\
&\quad (\neg P1 \wedge P2 \wedge P3 \wedge \neg P4) \vee\\
&\quad (\neg P1 \wedge P2 \wedge \neg P3 \wedge P4) \vee\\
&\quad (\neg P1 \wedge \neg P2 \wedge P3 \wedge P4)\\
&) \wedge\\
&\neg(P1 \wedge P2) \wedge \neg(P1 \wedge P3) \wedge\\
&\neg(P1 \wedge P4) \wedge \neg(P2 \wedge P3) \wedge\\
&\neg(P2 \wedge P4) \wedge \neg(P3 \wedge P4)\\
&\leftrightarrow F
\end{aligned}
$$

Now we can use laws to convert the formula on the left to F. Try to use distributive law first.

Example

$$
\begin{aligned}
&\boxed{(P1 \wedge P2)} \wedge \neg P3 \wedge \neg P4 \wedge \\
&\boxed{\neg(P1 \wedge P2)} \wedge \neg(P1 \wedge P3) \wedge \\
&\neg(P1 \wedge P4) \wedge \neg(P2 \wedge P3) \wedge \\
&\neg(P2 \wedge P4) \wedge \neg(P3 \wedge P4) \vee \\
&\ldots\ldots \vee \\
&\neg P1 \wedge \neg P2 \wedge \boxed{(P3 \wedge P4)} \wedge \\
&\neg(P1 \wedge P2) \wedge \neg(P1 \wedge P3) \wedge \\
&\neg(P1 \wedge P4) \wedge \neg(P2 \wedge P3) \wedge \\
&\neg(P2 \wedge P4) \wedge \boxed{\neg(P3 \wedge P4)}
\end{aligned}
$$

Now commutative law
and associative law.

Example

$$\boxed{(P1 \wedge P2) \wedge \neg(P1 \wedge P2)} \wedge$$
$$\neg P3 \wedge \neg P4 \wedge \neg(P1 \wedge P3) \wedge$$
$$\neg(P1 \wedge P4) \wedge \neg(P2 \wedge P3) \wedge$$
$$\neg(P2 \wedge P4) \wedge \neg(P3 \wedge P4) \vee$$
$$\ldots\ldots \vee$$
$$\boxed{\neg(P3 \wedge P4) \wedge (P3 \wedge P4)} \wedge$$
$$\neg(P1 \wedge P2) \wedge \neg(P1 \wedge P3) \wedge$$
$$\neg(P1 \wedge P4) \wedge \neg(P2 \wedge P3) \wedge$$
$$\neg(P2 \wedge P4) \wedge \neg P1 \wedge \neg P2$$

Now Complementary law.

Example

$$F \vee F \vee F \vee F \vee F \vee F$$

$$\Downarrow$$

$$F$$

$$F \leftrightarrow F$$

We have successfully convert the left to F.

Exercise

$$P \to Q = \neg Q \to \neg P$$

$$P \to (Q \to R) = Q \to (P \to R)$$

$$P \to (Q \to R) = (P \wedge Q) \to R$$

Can you prove them by laws instead of truth table?