More Propositional Logic Algebra: Expressive Completeness and Completeness of Equivalences

Equivalences Involving Conditionals

Some Important Equivalences Involving Conditionals

- Implication:
  - $P \rightarrow Q \Leftrightarrow \neg P \vee Q$
  - $\neg(P \rightarrow Q) \Leftrightarrow P \wedge \neg Q$
- Contraposition (or Transposition):
  - $P \rightarrow Q \Leftrightarrow \neg Q \rightarrow \neg P$
- Exportation:
  - $P \rightarrow (Q \rightarrow R) \Leftrightarrow (P \wedge Q) \rightarrow R$
- Equivalence:
  - $P \leftrightarrow Q \Leftrightarrow (P \rightarrow Q) \wedge (Q \rightarrow P)$
  - $P \leftrightarrow Q \Leftrightarrow (P \wedge Q) \vee (\neg P \wedge \neg Q)$

Some More Equivalences

- Distribution:
  - $P \rightarrow (Q \wedge R) \Leftrightarrow (P \rightarrow Q) \wedge (P \rightarrow R)$
  - $P \rightarrow (Q \vee R) \Leftrightarrow (P \rightarrow Q) \vee (P \rightarrow R)$
  - $(P \vee Q) \rightarrow R \Leftrightarrow (P \rightarrow R) \wedge (Q \rightarrow R)$
  - $(P \wedge Q) \rightarrow R \Leftrightarrow (P \rightarrow R) \vee (Q \rightarrow R)$ (this last one is a good example of the paradox of material implication!)

- Conditional Reduction:
  - $(P \rightarrow Q) \wedge P \Leftrightarrow P \wedge Q$
  - $(P \rightarrow Q) \wedge \neg Q \Leftrightarrow \neg P \wedge \neg Q$

Also:

$P \rightarrow P \Leftrightarrow \top$

$P \rightarrow \neg P \Leftrightarrow \neg P$

$P \rightarrow \bot \Leftrightarrow \neg P$

$P \rightarrow \top \Leftrightarrow \top$

$\top \rightarrow P \Leftrightarrow P$

$\bot \rightarrow P \Leftrightarrow \top$

Normal Forms

Negation Normal Form

- *Literals*: Atomic Sentences or negations thereof.
- *Negation Normal Form*: An expression built up with '$\wedge$', '$\vee$', and literals.
- Using repeated DeMorgan and Double Negation, we can transform *any* truth-functional expression built up with '$\wedge$', '$\vee$', and '$\neg$' into an expression that is in Negation Normal Form.
- Example:

$\neg((A \vee B) \wedge \neg C) \Leftrightarrow$ (DeMorgan)

$\neg(A \vee B) \vee \neg\neg C \Leftrightarrow$ (Double Neg, DeM)

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

Disjunctive Normal Form

- *Disjunctive Normal Form*: A generalized disjunction of generalized conjunctions of literals.
- Using repeated distribution of $\wedge$ over $\vee$, *any* statement in Negation Normal Form can be written in Disjunctive Normal Form.
- Example:

$(A \vee B) \wedge (C \vee D) \Leftrightarrow$ (Distribution)

$[(A \vee B) \wedge C] \vee [(A \vee B) \wedge D] \Leftrightarrow$ (Distribution (2x))

$(A \wedge C) \vee (B \wedge C) \vee (A \wedge D) \vee (B \wedge D)$

Conjunctive Normal Form

- *Conjunctive Normal Form*: A generalized conjunction of generalized disjunctions of literals.
- Using repeated distribution of $\vee$ over $\wedge$, *any* statement in Negation Normal Form can be written in Conjunctive Normal Form.
- Example:

$(A \wedge B) \vee (C \wedge D) \Leftrightarrow$ (Distribution)

$[(A \wedge B) \vee C] \wedge [(A \wedge B) \vee D] \Leftrightarrow$ (Distribution (2x))

$(A \vee C) \wedge (B \vee C) \wedge (A \vee D) \wedge (B \vee D)$

Special Cases

- Any literal (such as A or $\neg$B) is in NNF, DNF (it is a disjunction whose only disjunct is a conjunction whose only conjunct is that literal), and CNF
- A conjunction of literals (e.g. $\neg A \wedge \neg B \wedge C$) is in NNF, DNF (a disjunction whose only disjunct is that conjunction), and CNF
- Likewise, a disjunction of literals is in NNF, DNF, and CNF
- In particular, $\top$ and $\bot$ are in NNF, DNF, and CNF as well.

Expressive Completeness

Truth-Functional Connectives

- So far, we have seen one *unary* truth-functional connective (‘$\neg$’), and four *binary* truth-functional connectives (‘$\wedge$’, ‘$\vee$’, ‘$\rightarrow$’, ‘$\leftrightarrow$’).
- However, there are many more truth-functional connectives possible:
  - First of all, a connective can take any number of arguments: 3 (ternary), 4, 5, etc.
  - Second, there are unary and binary connectives other than the ones listed above.

Unary Connectives

- What other unary connectives are there besides '¬'?
- Thinking about this in terms of truth tables, we see that there are 4 different unary connectives:

P	*P
T	T
F	T

P	*P
T	T
F	F

P	*P
T	F
F	T

P	*P
T	F
F	F

Binary Connectives

- The truth table below shows that there are $2^4 = 16$ binary connectives:

P	Q	P*Q
T	T	T/F
T	F	T/F
F	T	T/F
F	F	T/F

In general:
n sentences $\Rightarrow$

2^n truth value combinations
(i.e. 2^n rows in truth table) $\Rightarrow$

2^{2^n} different n-ary connectives!

Expressing other connectives using ‘and’, ‘or’, and ‘not’

- We saw that we can express the exclusive disjunction using ‘and’, ‘or’, and ‘not’.
- Q: Can we express all other connectives as well?
- A: Yes! We can generalize from this example:

P	Q	P*Q	Step 1:	Step 2:
T	T	F		
T	F	T	$\Rightarrow P \wedge \neg Q$	$\Rightarrow (P \wedge \neg Q) \vee (\neg P \wedge Q)$
F	T	T	$\Rightarrow \neg P \wedge Q$	
F	F	F		

Truth-Functional Expressive Completeness

- *Any* expression using *any* truth-functional operators can be rewritten as a Boolean expression, i.e. an expression that only uses $\wedge$'s, $\vee$'s, and $\neg$'s.
- Since I can express *any* truth function using '$\wedge$', '$\vee$', and '$\neg$', we say that the set of operators $\{\neg, \wedge, \vee\}$ is (truth-functionally) *expressively complete*.

$\{\neg, \wedge\}$ and $\{\neg, \vee\}$ are Expressively Complete

- Using DeMorgan Laws and Double Negation:
  - $P \wedge Q \Leftrightarrow \neg(\neg P \vee \neg Q)$
  - $P \vee Q \Leftrightarrow \neg(\neg P \wedge \neg Q)$
- Hence, by the principle of substitution of logical equivalents, since $\{\neg, \wedge, \vee\}$ is expressively complete, the sets $\{\neg, \wedge\}$ and $\{\neg, \vee\}$ are expressively complete as well!

The NAND

- Let us define the binary truth-functional connective ‘NAND’ according to the truth-table below.
- Obviously, P NAND Q $\Leftrightarrow \neg(P \wedge Q)$ (hence the name!)

P	Q	P NAND Q
T	T	F
T	F	T
F	T	T
F	F	T

Expressive Completeness of the NAND

- The NAND is very interesting, because the {NAND} is expressively complete!
- Proof: We already know that we can express every truth-functional connective using only $\vee$ and $\neg$. Furthermore:
  - P NAND P $\Leftrightarrow \neg(P \wedge P) \Leftrightarrow \neg P$
  - (P NAND P) NAND (Q NAND Q) $\Leftrightarrow \neg$((P NAND P) $\wedge$ (Q NAND Q)) $\Leftrightarrow \neg(\neg P \wedge \neg Q) \Leftrightarrow P \vee Q$

The NOR

- Let us define the binary truth-functional connective ‘NOR’ according to the truth-table below.
- Obviously, P NOR Q $\Leftrightarrow \neg(P \vee Q)$ (hence the name!)

P	Q	P NOR Q
T	T	F
T	F	F
F	T	F
F	F	T

Expressive Completeness of the NOR

- Like the NAND, the NOR can express any truth-functional connective, i.e. {NOR} is expressively complete as well!
- Proof: We already know that we can express every truth-functional connective using only $\wedge$ and $\neg$. Furthermore:
  - P NOR P $\Leftrightarrow \neg(P \vee P) \Leftrightarrow \neg P$
  - (P NOR P) NOR (Q NOR Q) $\Leftrightarrow \neg(($P NOR P$) \vee ($Q NOR Q$)) \Leftrightarrow \neg(\neg P \vee \neg Q) \Leftrightarrow P \wedge Q$

Completeness of Equivalence Rules

Completeness of Equivalence Rules

- If we regard some set S of equivalence principles as formal, syntactical, rewriting principles, then we can define:
  - $\varphi \leftrightarrow_S \psi$ iff through the successive use of equivalence principles, φ can be rewritten into ψ
- S is a *complete* set of equivalence rules iff:
  - For any φ and ψ: if $\varphi \Leftrightarrow \psi$ then $\varphi \leftrightarrow_S \psi$

How to Prove Completeness?

- I claim that the following set BS of equivalence rules is complete (restricting ourselves to statements involving Boolean operators only):
  - Commutation
  - Association
  - Double Negation
  - DeMorgan
  - Distribution
  - Idempotence
  - Adjacency
  - Identity
  - Inverse
  - Complement
  - Annihilation
- How do I prove this?

Proof Using Normal Forms

- Let's say that the canonical form of a statement φ is the statement $CF(\varphi)$ that is the statement you get from using the 'truth-table' trick to get a statement's equivalent expression:
  - Reference columns followed pre-set ordering of atomic statements
  - Reference columns are filled out using pre-set alternation scheme
  - Conjuncts are conjuncted following same alphabetical order
  - Disjuncts are disjuncted from top row to bottom
- Theorem: For any φ: $\varphi \leftrightarrow_{BS} CF(\varphi)$
  - And hence, for any φ and ψ: if $\varphi \Leftrightarrow \psi$ then $\varphi \leftrightarrow_{BS} CF(\varphi) = (!)\ CF(\psi) \leftrightarrow_{BS} \psi$