Propositional Logic – Syntax – Part 2

Atomic formula: A formula is atomic if it is a propositional variable or it is a True (T) or False (F) value.

Propositional formula

- A propositional formula can be consist of:
  - Propositional variables: p, q, r...,
  - Unary connectives,¬
  - Binary connectives ∧,∨, ⇒, ⇔
  - Parentheses (and).
- **Well-formed formulas (WFF):** The well-formed formulas of propositional logic are obtained by using the construction rules:

1. The atomic formula p is a well-formed formula,
2. If P is a well-formed formula, then so is $\neg P$.
3. If P and Q are a well-formed formula, then so are $P \wedge Q, P \vee Q,\ P \Rightarrow Q, and\ P \Leftrightarrow Q$
4. If P is a well-formed formula, then so is (P).

- By using bellow four rules, we can check that a formula is a well-formed formula.
- **Example 1:** Are the following formulas well formed?
  - $\boldsymbol{p}$ is an atomic variable, so is a WFF. (R1)
  - $p \Rightarrow q$ is a WFF. (R1,R3)
  - $(\boldsymbol{p}\neg\boldsymbol{q}) \Rightarrow \boldsymbol{p}$ is not a WFF (does not respect R3)
  - $\neg(\wedge\ \boldsymbol{Q})$ is not a WFF (does not respect R2 and R3)
- **Example 2:** $\big((P \vee \neg Q) \Rightarrow (\neg(P \Leftrightarrow Q))\big)$

Parsing tree:

- We use the parsing tree to check that a formula is WFF.
- A formula can be represented as a tree.
- Internal nodes are the logical connectives,
- Logical connectives (∧,∨, ⇒, ⇔) are binary nodes,

- The negation (,¬) is a unary node,
- The leaves of the tree are atomic variables.
- The construction of this tree is based on writing rules.
- This tree can be read from bottom to top (composition of the formula) or from top to bottom (decomposition of the formula).
- There is only one (unique) parsing tree for each formula.
    - **Example:** Create the construction tree of this formula $\big((P \vee \neg Q) \Rightarrow (\neg(P \Leftrightarrow Q))\big)$
- We say that two formulas ($P = Q$) are syntactically equal if only if these formulas represented by the same tree, the name of the atomic variables can be different.
    - **Example:** $p \vee q$=a $\vee$ b

Depth of a formula:

- The depth of a formula F corresponds to the depth of the parsing tree (i.e. the number of branches separating the root from the furthest leaf).

Formula	Depth
A	0
¬(A)	1
¬((A)∧(B))	2
(¬(A))∧(B)	2
((A)∧(B))→(C)	2
((A)∧(B))→((C)∧(D))	2

Complexity of a formula

- The complexity of a formula counts the number of connectives.

Formula	Complexity
A	0
¬(A)	1
¬((A)∧(B))	2
(¬(A))∧(B)	2
((A)∧(B))→(C)	2
((A)∧(B))→((C)∧(D))	3

Subformula:

- The set of subformulas of a formula F is the smallest set.
- Example: Using the parsing tree to find the set of subformulas of $(\boldsymbol{p} \vee \boldsymbol{q}) \Rightarrow \neg \boldsymbol{p}$ is: {p, q , $\neg p$, $p \vee q$, $(p \vee q) \Rightarrow \neg p$}.

Simplifying a formula

- The number of parentheses sometimes makes writing propositional forms complex.
- To make writing PF easier, we remove certain superfluous parentheses, taking into account certain conventions:
  - Removing parentheses placed on either side of a propositional variable. Example: (p) becomes p, $((P) \vee (Q)) \Rightarrow (P)$ ➔$(P \vee Q) \Rightarrow P$
  - Removed parentheses separating consecutive negations. Example: ٦(٦(p)) becomes ٦٦p.
  - Apply the priority rules between connectives.

Substitution in a formula:

- A substitution associates a formula A with a propositional variable p.
- It is noted [p\A].
- The application of [p\A] to a formula B, denoted (B)[p\A], is the result of replacing all occurrences of p in B by A.
- **Examples:**
  - $(p \Rightarrow q)[p \backslash r] = r \Rightarrow q$
  - $(p \vee q)[p \backslash q] = q \vee q$
  - $((p \vee q) \Rightarrow \neg p)[p \backslash \neg p] = (\neg p \vee q) \Rightarrow \neg \neg p$
  - $((p \vee q) \Rightarrow \neg p)[p \backslash (p \wedge r)] =$, the location of the parentheses is very important$((p \wedge r) \vee q) \Rightarrow \neg (p \wedge r)$

Simultaneous substitution

- The last definition can be easily generalized to a simultaneous substitution (B)[p1\A1, …, pn\An,], such that p1,…,pn are all different variables.
- Example :

- $((p \vee q) \Rightarrow \neg p)[p \backslash \neg p, q \backslash (r \wedge p)] = (\neg p \vee (r \wedge p)) \Rightarrow \neg\neg p$
- **Note:** the results of simultaneous substitution and sequential substitution are different.
- Example:
  - Case 1: $((p \vee q) \Rightarrow \neg p)[p \backslash p \wedge q, q \backslash r \wedge p] = \big((\boldsymbol{p} \wedge \boldsymbol{q}) \vee (\boldsymbol{r} \wedge \boldsymbol{p})\big) \Rightarrow \neg(\boldsymbol{p} \wedge \boldsymbol{q})$
- Case 2:
  - $((p \vee q) \Rightarrow \neg p)[p \backslash p \wedge q] = ((p \wedge q) \vee q) \Rightarrow \neg p(p \wedge q)$
  - $((p \vee q) \Rightarrow \neg p)[q \backslash r \wedge p] = ((\boldsymbol{p} \wedge (\boldsymbol{r} \wedge \boldsymbol{p})) \vee (\boldsymbol{r} \wedge \boldsymbol{p})) \Rightarrow \neg \boldsymbol{p}(\boldsymbol{p} \wedge (\boldsymbol{r} \wedge \boldsymbol{p}))$

Exercise 2: Check that the following expressions are well-formed formulas using the parsing tree.

- $((p_1 \Leftrightarrow p_2) \vee p_3) \wedge (p_2 \wedge (p_3 \Rightarrow p_4))$
- $p_1 \wedge (p_2 = p_3)$