Propositional Logic

Motivation for studying Logic: To acquire the ability to model real-life situations in a way that would allow us to reason about them formally.

Example 1: If the train arrives late and there are no taxis at the station, then John is late for his meeting. John is not late for his meeting. The train did arrive late. Therefore, there were taxis at the station.

Example 2: If it is raining and Jane does not have her umbrella with her, then she will get wet. Jane is not wet. It is raining. Therefore, Jane has her umbrella with her.

Can we verify the validity of these arguments formally?

- We need to turn the English sentences into formulas (modeling).
- Then, we can apply mathematical reasoning to formulas.

Modelling

Encoding:

<table>
<thead>
<tr>
<th>Example 1</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>p</td>
</tr>
<tr>
<td>the train is late</td>
<td>it is raining</td>
</tr>
<tr>
<td>q</td>
<td>q</td>
</tr>
<tr>
<td>there are taxis at the station</td>
<td>June has her umbrella with her</td>
</tr>
<tr>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td>John is late for his meeting</td>
<td>Jane gets wet</td>
</tr>
</tbody>
</table>

Patterns:

If $p$ and not $q$, then $r$. Not $r$, $p$. Therefore $q$.

We shall study reasoning patterns.

Declarative Sentences

Declarative sentences (we can consider whether they’re true or not):

- The sum of the numbers 3 and 5 equals 8.
- Jane reacted violently to Jack’s accusations.
- Every even natural number is the sum of two prime numbers.
- All Martians like pepperoni on their pizzas.

Non-declarative sentences (can’t tell whether they’re true or not):

- Could you please pass the salt.
- Ready, steady, go.
- May fortune come your way.

We want to turn declarative sentences into formulas and create a formalism to manipulate such formulas.

Turning English Phrases into Formulas

Atomic sentences:

- $p$: I won the lottery last week.
- $q$: I purchased a lottery ticket.
- $r$: I won last week’s sweepstakes.

Connectives:

- $\neg$: negation — $\neg p$: I did not win the lottery.
- $\lor$: disjunction — $p \lor q$: I won the lottery last week or I won last week’s sweepstakes.
- $\land$: conjunction — $p \land r$: I won the lottery and the sweepstakes last week.
- $\rightarrow$: implication — $p \rightarrow q$: If I won the lottery last week, then I purchased a lottery ticket.

Composite formulas: $p \rightarrow (q \land p)$: connective priority; $\neg \land \lor \rightarrow$.

By this convention, we can remove the brackets: $p \rightarrow q \rightarrow (\neg \land \lor)$.

Natural Deduction

- Collection of proof rules, which allow to infer new formulas from existing formulas.
- Given the formulas $\Phi_1, \ldots, \Phi_n$, we intend to infer a conclusion $\Psi$. We denote this by $\Phi_1, \ldots, \Phi_n \rightarrow \Psi$.
- This construct is called a sequent.
- Example: $p \land q \rightarrow r \rightarrow p \lor q$.

- There is no "perfect" set of proof rules. You can create your own (you can even invent your own logic). Such exercise resembles computer programming.
Natural Deduction — Derived Rules

\[
\frac{\Phi \rightarrow \Psi \rightarrow \Phi \rightarrow \neg\Phi}{\text{MT}}
\]

1. \(\Phi \rightarrow \Psi\) premise
2. \(\neg\Psi\) premise
3. \(\Phi\) assumption
4. \(\neg\Phi\) RAA (Reductio Ad Absurdum)

Justification: If I am Chinese, then I am Asian. I am not Asian, Therefore, I'm not Chinese.

Natural Deduction Summary

Basic rules:

<table>
<thead>
<tr>
<th>(\phi)</th>
<th>(\psi)</th>
<th>(\chi)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\phi \rightarrow \psi)</td>
<td>(\psi \rightarrow \chi)</td>
<td>(\phi \rightarrow (\psi \rightarrow \chi))</td>
</tr>
<tr>
<td>(\phi \rightarrow (\psi \rightarrow \chi))</td>
<td>(\psi \rightarrow (\phi \rightarrow \chi))</td>
<td>(\chi \rightarrow (\phi \rightarrow \psi))</td>
</tr>
<tr>
<td>(\phi \rightarrow (\psi \rightarrow \chi))</td>
<td>(\psi \rightarrow (\phi \rightarrow \chi))</td>
<td>(\chi \rightarrow (\phi \rightarrow \psi))</td>
</tr>
</tbody>
</table>

Useful derived rules:

\[
\frac{\phi \rightarrow \chi \rightarrow \psi}{\text{Sub}}
\]

Propositional Logic as a Formal Language

Proofs are in fact proof schemes:

\[
p \rightarrow q, p \vdash q\]

1. \(p \rightarrow q\) premise
2. \(p \vdash r\) premise
3. \(q \rightarrow s\) assumption
4. \(r \vdash s\) premise
5. \(s \rightarrow r\) given
6. \(r \vdash s\) given
7. \(s \rightarrow r\) given
8. \(r \vdash s\) given
9. \(s \rightarrow r\) given
10. \(r \vdash s\) given
11. \(s \rightarrow r\) given
12. \(r \vdash s\) given
13. \(s \rightarrow r\) given
14. \(r \vdash s\) given
15. \(s \rightarrow r\) given
16. \(r \vdash s\) given
17. \(s \rightarrow r\) given
18. \(r \vdash s\) given
19. \(s \rightarrow r\) given
20. \(r \vdash s\) given
21. \(s \rightarrow r\) given
22. \(r \vdash s\) given
23. \(s \rightarrow r\) given
24. \(r \vdash s\) given
25. \(s \rightarrow r\) given
26. \(r \vdash s\) given
27. \(s \rightarrow r\) given
28. \(r \vdash s\) given
29. \(s \rightarrow r\) given
30. \(r \vdash s\) given
31. \(s \rightarrow r\) given
32. \(r \vdash s\) given
33. \(s \rightarrow r\) given
34. \(r \vdash s\) given
35. \(s \rightarrow r\) given
36. \(r \vdash s\) given
37. \(s \rightarrow r\) given
38. \(r \vdash s\) given
39. \(s \rightarrow r\) given
40. \(r \vdash s\) given
41. \(s \rightarrow r\) given
42. \(r \vdash s\) given
43. \(s \rightarrow r\) given
44. \(r \vdash s\) given
45. \(s \rightarrow r\) given
46. \(r \vdash s\) given
47. \(s \rightarrow r\) given
48. \(r \vdash s\) given
49. \(s \rightarrow r\) given
50. \(r \vdash s\) given
51. \(s \rightarrow r\) given
52. \(r \vdash s\) given
53. \(s \rightarrow r\) given
54. \(r \vdash s\) given
55. \(s \rightarrow r\) given
56. \(r \vdash s\) given
57. \(s \rightarrow r\) given
58. \(r \vdash s\) given
59. \(s \rightarrow r\) given
60. \(r \vdash s\) given
61. \(s \rightarrow r\) given
62. \(r \vdash s\) given
63. \(s \rightarrow r\) given
64. \(r \vdash s\) given
65. \(s \rightarrow r\) given
66. \(r \vdash s\) given
67. \(s \rightarrow r\) given
68. \(r \vdash s\) given
69. \(s \rightarrow r\) given
70. \(r \vdash s\) given
71. \(s \rightarrow r\) given
72. \(r \vdash s\) given
73. \(s \rightarrow r\) given
74. \(r \vdash s\) given
75. \(s \rightarrow r\) given
76. \(r \vdash s\) given
77. \(s \rightarrow r\) given
78. \(r \vdash s\) given
79. \(s \rightarrow r\) given
80. \(r \vdash s\) given
81. \(s \rightarrow r\) given
82. \(r \vdash s\) given
83. \(s \rightarrow r\) given
84. \(r \vdash s\) given
85. \(s \rightarrow r\) given
86. \(r \vdash s\) given
87. \(s \rightarrow r\) given
88. \(r \vdash s\) given
89. \(s \rightarrow r\) given
90. \(r \vdash s\) given
91. \(s \rightarrow r\) given
92. \(r \vdash s\) given
93. \(s \rightarrow r\) given
94. \(r \vdash s\) given
95. \(s \rightarrow r\) given
96. \(r \vdash s\) given
97. \(s \rightarrow r\) given
98. \(r \vdash s\) given
99. \(s \rightarrow r\) given
100. \(r \vdash s\) given

Syntax Trees

Well-formed formula: \((p \rightarrow q) \rightarrow (p \rightarrow (q \rightarrow -q)))

Semantics of Propositional Logic — Truth Values

The semantics of propositional logic is a mapping

\[
\text{Interpretation : } \text{WFF} \rightarrow \{T,F\}
\]

where \(T\) stands for true and \(F\) stands for false. The semantics has to be consistent with the connectives \(\neg, \land, \lor, \rightarrow\). This consistency is specified by the following truth table.

<table>
<thead>
<tr>
<th>(\Phi)</th>
<th>(\Psi)</th>
<th>(\neg\Phi)</th>
<th>(\Phi \land \Psi)</th>
<th>(\Phi \lor \Psi)</th>
<th>(\neg (\neg \Phi))</th>
<th>(\Phi \rightarrow \Psi)</th>
<th>(\Psi \rightarrow \Phi)</th>
<th>(\Phi \rightarrow T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Truth tables are means of exploring all possible interpretations for a given formula.
Truth Table Example

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>r</th>
<th>p \land q \rightarrow p \land (q \lor \neg r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

Semantics of Propositional Logic — Sequents

Given a sequent \( \Phi_1, \Phi_2, \ldots, \Phi_n \vdash \Psi \) (which we don’t know whether it is valid), we denote by

\[ \Phi_1, \Phi_2, \ldots, \Phi_i \vdash \Psi \]

a new kind of sequent, which is valid if for every semantics \( S \) such that \( S(\Phi_i) = T \) \( i = 1 \ldots n \), we also have that \( S(\Psi) = T \). The \( \vdash \) relation is called semantic entailment.

Example: \( p \rightarrow q \rightarrow r \)

### Intermezzo — Mathematical Induction

How do we prove that \( 1 + 2 + \cdots + n = \frac{n(n+1)}{2} \)? **Answer:** Mathematical induction.

(Base case) We prove the statement for \( n = 1 \). Indeed, \( 1 = \frac{1(1+1)}{2} \).

(Induction case) We assume that the statement is true for some general value of \( n \), and we show that it implies the statement for \( n + 1 \). In other words, we prove that

\[
1 + 2 + \cdots + n + (n + 1) = \frac{n(n+1)}{2} + (n + 1) = \frac{(n+1)(n+2)}{2}
\]

Indeed

\[
1 + 2 + \cdots + n + (n + 1) = \frac{n(n+1)}{2} + (n + 1) = \frac{(n+1)(n+2)}{2}
\]

### General Mathematical Induction Principle

Given a statement \( \eta(n) \) that depends on a natural number \( n \), and whose validity we want to prove for all possible values of \( n \), we proceed in the following two steps:

- **Base case:** prove that \( \eta(1) \) holds.
- **Induction case:** prove that \( \eta(1) \land \eta(2) \land \cdots \land \eta(n) \rightarrow \eta(n+1) \) for all natural numbers \( n \). When proving such a statement, we call \( \eta(n) \) the induction hypothesis.
- These two conditions prove \( \eta(n) \) for all \( n \).

### Course of Values Induction

Given a statement \( \eta(n) \) that depends on a natural number \( n \), and whose validity we want to prove for all possible values of \( n \), we proceed in the following two steps:

- **Base case:** prove that \( \eta(1) \) holds.
- **Induction case:** prove that \( \eta(1) \land \eta(2) \land \cdots \land \eta(n) \rightarrow \eta(n+1) \) for all natural numbers \( n \). When proving such a statement, we call \( \eta(n) \) the induction hypothesis.
- These two conditions prove \( \eta(n) \) for all \( n \).

### Course of Values Induction Example

**Definition:** Given a well-formed formula \( \Phi \), we define its height to be 1 plus the length of its largest path of its parse tree.

**Theorem:** For every well-formed propositional logic formula, the number of left brackets is equal to the number of right brackets.

**Proof:** Denote by \( \eta(n) \) the statement "all formulas \( \Phi \) of height \( n \) have the same number of left and right brackets." 

- **Base case:** \( n = 1 \). \( \eta(1) \) applies to all propositional formulas \( p, q, \ldots \) and obviously holds.
- **Induction case:** \( n > 1 \). Then the root of the parse tree of \( \Phi \) is one of the connectives \( \land, \lor, \rightarrow \). We assume that it is \( \rightarrow \) (the other cases are proved in a similar manner). Then \( \Phi = \Phi_1 \rightarrow \Phi_2 \) for some wfs \( \Phi_1 \) and \( \Phi_2 \), whose heights are strictly smaller than \( n \). Using the induction hypothesis, the number of left and right brackets is equal for both \( \Phi_1 \) and \( \Phi_2 \). \( \Phi \) adds only two brackets, one \( \rightarrow \) and one \( \land \) or \( \lor \). Therefore, the statement is correct.
Soundness and Completeness of Propositional Logic

When we define a logic (or any type of calculus), we want to show that it is useful.

- **Soundness:** Formulas that we derive using the calculus reflect a "real" truth.
- **Completeness:** Every formula corresponding to a "real" truth can be inferred using the rules of the calculus.

In the case of propositional logic, given the wffs $\Phi_1, \Phi_2, \ldots, \Phi_n$ and $\Psi$, we have

- **Soundness:** if $\Phi_1, \ldots, \Phi_n \vdash \Psi$ holds, then $\Phi_1, \ldots, \Phi_n \models \Psi$ holds.
- **Completeness:** if $\Phi_1, \ldots, \Phi_n \models \Psi$ holds, then $\Phi_1, \ldots, \Phi_n \vdash \Psi$ holds.