Part Two: The Basic Components of the SOFL Specification Language

- SOFL logic
- Module
- Condition Data Flow Diagrams
- Process specification
- Function definition and specification
- Process decomposition
- Other issues
2. Propositional logic

SOFL logic is an extension of classical propositional logic and predicate logic; it allows “undefined” as a logical value (SOFL adopts the Jones’ three-value logic used in VDM).
Definition 2.1 A proposition is a statement that must be either true or false.

For example, the following statements are propositions:

(1) A tiger is an animal (true)
(2) An apple is a fruit (true)
(3) $3 + 5 > 10$ (false)
In contrast, the following statements are not propositions:

(1) Are you happy?

(2) Let's go swimming

(3) $x := y + 3$ (assignment statement)
Definition 2.2 The value `true` and `false` are called truth value.

In SOFL we use `bool` to represent the Boolean type that contains the truth values, that is:

\[
\text{bool} = \{\text{true}, \text{false}\}
\]

Propositions are represented by symbols:

(1) P: A tiger is an animal.
(2) Q: An apple is a fruit.
(3) R: 3 + 5 > 10.

Such a proposition is called atomic proposition (which cannot be decomposed).
Propositions can be connected using logical operators to form propositional expressions (or compound propositions) that describe more complicated propositions.
2.2 propositional operators:

<table>
<thead>
<tr>
<th>operator</th>
<th>read as</th>
<th>priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>not</td>
<td>not</td>
<td>highest</td>
</tr>
<tr>
<td>and</td>
<td>and</td>
<td></td>
</tr>
<tr>
<td>or</td>
<td>or</td>
<td></td>
</tr>
<tr>
<td>=&gt;</td>
<td>implies</td>
<td></td>
</tr>
<tr>
<td>&lt;=&gt;</td>
<td>is equivalent to</td>
<td>lowest</td>
</tr>
</tbody>
</table>
2.3 Conjunction

Definition 2.3 A conjunction is a propositional expression whose principal operator is \textit{and}.

For example:

\[
x > 5 \text{ and } x < 10
\]

Question: How to decide the truth value of a conjunction?
### Truth table for conjunction

<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>P1 and P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

**Examples:**

- true and true $\iff$ true
- false and true $\iff$ false
- false and false $\iff$ false
2.4 Disjunction

Definition 2.4 A disjunction is a propositional expression whose principal operator is or.

\[ P_1 \text{ or } P_2 \]

For example:

\[ x > 5 \text{ or } x < 3 \]
<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>P1 or P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>
2.5 Negation

Definition 2.5 A negation is a propositional expression whose principal operator is not.

Example:

\[ \text{not } P_1 \]

\[ \text{not } x > 5 \]
<table>
<thead>
<tr>
<th>P1</th>
<th>not P1</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>
2.6 Implication

Definition 2.6 An implication is a propositional expression whose principal operator is $\implies$.

$P_1 \implies P_2$
<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>P1 =&gt; P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>

Example:

\[ x > 10 \implies x > 5 \]

In this case we can also say that \( x > 10 \) is stronger than \( x > 5 \).
2.7 Equivalence

Definition 2.7 An equivalence is a propositional expression whose principal operator is $\iff$.

$$P_1 \iff P_2$$
<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>P1 $\iff$ P2</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
</tr>
</tbody>
</table>

Examples:

(1) John is a friend of Chris $\iff$ Chris is a friend of John
(2) $x > 10 \iff$ not $x = 10$ and not $x < 10$
The use of parenthesis

An expression is interpreted by applying the operator priority order unless parenthesis is used.

For example: the expression

\[ \text{not } p \text{ and } q \text{ or } r \iff p \implies q \text{ and } r \]

is equivalent to the expression:

\[ (((\text{not } p) \text{ and } q) \text{ or } r) \iff (p \implies (q \text{ and } r)) \]

Parenthesis can be used to change the precedence of operators in expressions. For example, the above expression can be changed to:

\[ \text{not } (p \text{ and } ((q \text{ or } (r \iff p)) \implies q) \text{ and } r) \]
2.8 Tautology, contradiction, and contingency

Definition 2.8 A tautology is a proposition that evaluates as true in every combination of the truth values of its constituent propositions.

Examples:

(1) \( P \) or not \( P \)
(2) \( x > 10 \) or \( x \leq 10 \)
Definition 2.9 A contradiction is a proposition that evaluates as false in every combination of the truth values of its constituent propositions.

In other words, a contradiction is a negation of a tautology.

Examples:

(1) P and not P
(2) x > 10 and x < 10
Definition 2.10 A contingency is a propositional expression that is neither a tautology nor a contradiction.

In other words, a contingency can evaluate as both true and false, respectively.

Examples:

(1) P and Q  (P and Q are not related with each other)

(2) x > 5 or x < -5
2.9 Normal forms

Definition 2.11 A disjunctive normal form is a special kind of disjunction in which each constituent propositional expression, usually known as disjunctive clause, is a conjunction of atomic propositions or their negations.

Example:

\( P_1 \text{ or } P_2 \text{ or } \cdots \text{ or } P_n \)

where \( P_i \) (i=1..n) is a conjunction of atomic propositions or their negations.

Characteristics of a disjunctive normal form:

It evaluates as true as long as one of the disjunctive clauses evaluates as true.
Definition 2.12 A conjunctive normal form is a special kind of conjunction in which each constituent propositional expression, usually called conjunctive clause, is a disjunction of atomic propositions or their negations.

Example:

\[ Q_1 \land Q_2 \land \cdots \land Q_m \]

where \( Q_j \ (j=1..m) \) is a disjunction of atomic propositions or their negations.

Characteristics of a conjunctive normal form:
It evaluates as false as long as one of its conjunctive clauses evaluates as false.
2.10 Sequent

Definition 2.13 A sequent is an assertion that a conclusion can be deduced from hypotheses.

Let \( P_1, P_2, \ldots, P_n \) be hypotheses (propositional expressions) and \( Q \) a conclusion. Then a sequent is written as:

\[
P_1, P_2, \ldots, P_n \vdash Q
\]

where \( \vdash \) is called a turnstile.
Examples:

(1) P and Q ⊢ P

(2) P => Q, P ⊢ Q

A sequent can be used to represent a theorem, property, or a judgement.
2.11 Proof

Definition 2.14 A proof is a process (or activity or evidence) to show that the conclusion can be established from its hypotheses in a sequent.

Two methods for proof:
(1) By truth table
(2) By inference (推理) (specifically, natural deduction(演えき))

Example: to give a proof for the sequent:

\[ \neg P \text{ and } Q \vdash P \Rightarrow Q \]

we can build a truth table:
A truth table proof

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>not P and Q</th>
<th>P =&gt; Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>-</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>-</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>-</td>
</tr>
</tbody>
</table>
2.12 Inference rules

Natural deduction for proof needs appropriate inference rules. In this section, we introduce the necessary rules for proof in the propositional logic.

Proof strategies:

(1) forward proof

P₁, P₂, ..., Pₙ ⊢ Q

(2) proof by contradiction

P₁, P₂, ..., Pₙ, not Q ⊢ false
An inference rule is represented in the form:

1. \( \text{premise}_1, \text{premise}_2, \ldots, \text{premise}_n \)
   \( \begin{array}{c}
   \hline
   \text{conclusion}
   \end{array} \)
   \text{name}

2. \( \text{premise}_1, \text{premise}_2, \ldots, \text{premise}_n \)
   \( \begin{array}{c}
   \hline
   \text{conclusion}
   \end{array} \)
   \text{name}

Representation 1 means that the conclusion can be derived from \( \text{premise}_1, \ldots, \text{and premise}_n \), while representation 2 means that the conclusion and the group of premises can be derived from each other.
2.12.1 Rules for conjunction

P, Q

\[
\frac{P \text{ and } Q}{P} \quad \text{[and-intro]}
\]

P, Q

\[
\frac{P \text{ and } Q}{Q} \quad \text{[and-intro]}
\]

P, Q

\[
\frac{P \text{ and } Q}{P} \quad \text{[and-elim1]}
\]

P, Q

\[
\frac{P \text{ and } Q}{Q} \quad \text{[and-elim2]}
\]
2.12.2 Rules for disjunction

\[
\begin{align*}
P & \\
P \quad [\text{or-intro1}] & \\
P \lor Q & \\
Q & \\
Q \quad [\text{or-intro2}] & \\
P \lor Q & \\
P \lor Q, P \vdash R, Q \vdash R & \\
R & [\text{or-elim}]
\end{align*}
\]
2.12.3 Rules for negation

\[
\begin{align*}
P & \quad \text{[not-intro]} \\
\text{not not } P & \quad \text{[not-intro]} \\
\text{not not } P & \quad \text{[not-elim]} \\
P & \quad \text{[not-elim]}
\end{align*}
\]
2.12.4 Rules for implication

\[ \frac{Q}{P \Rightarrow Q} \quad [\Rightarrow\text{-intro}] \]

\[ P \Rightarrow Q \]

\[ \frac{P \Rightarrow Q, P}{Q} \quad [\Rightarrow\text{-elim}] \]
2.12.5 Rules for equivalence

\[
P \implies Q, \quad Q \implies P
\]

\[
\frac{P \iff Q}{P \implies Q}[\iff \text{-intro}]
\]

\[
P \iff Q
\]

\[
\frac{P \iff Q}{P \implies Q}[\iff \text{-elim1}]
\]

\[
P \implies Q
\]

\[
\frac{P \iff Q}{Q \implies P}[\iff \text{-elim2}]
\]
2.12.6 Properties of propositional expressions

(1) A conjunction, disjunction, or equivalence is commutative:

\[ P \text{ and } Q \]
\[ \underline{Q \text{ and } P} \] \[\text{[and-comm]}\]

\[ P \text{ or } Q \]
\[ \underline{Q \text{ or } P} \] \[\text{[or-comm]}\]

\[ P \iff Q \]
\[ \underline{Q \iff P} \] \[\text{[\iff-comm]}\]
(2) A conjunction, disjunction, implication, or equivalence is associative:

\[
P \land (Q \land R) = (P \land Q) \land R \quad \text{[and-ass]}
\]

\[
P \lor (Q \lor R) = (P \lor Q) \lor R \quad \text{[or-ass]}
\]

\[
P \Rightarrow (Q \Rightarrow R) = (P \Rightarrow Q) \Rightarrow R \quad \text{[=>-ass]}
\]

\[
P \Leftrightarrow (Q \Leftrightarrow R) = (P \Leftrightarrow Q) \Leftrightarrow R \quad \text{[<=>-ass]}
\]
(3) Conjunctions and disjunctions are distributive over each other:

\[
\begin{align*}
P \text{ and } (Q \text{ or } R) & \quad \text{[and-or-dist]} \\
(P \text{ and } Q) \text{ or } (P \text{ and } R) & \\
\hline
\end{align*}
\]

\[
\begin{align*}
P \text{ or } (Q \text{ and } R) & \quad \text{[or-and-dist]} \\
(P \text{ or } Q) \text{ and } (P \text{ or } R) & \\
\hline
\end{align*}
\]
(4) An implication is equivalent to a disjunction:

\[
P \implies Q \quad \text{[=>-or-equiv]}
\]
not P or Q
(5) Negations, conjunctions, and disjunctions satisfy the de Morgan's laws:

\[
\text{not} (P \text{ and } Q) \quad \quad \quad \quad \quad [\text{and-deM}]
\]
\[
\text{not} P \text{ or } \text{not} Q
\]

\[
\text{not} (P \text{ or } Q) \quad \quad \quad \quad \quad \quad [\text{or-deM}]
\]
\[
\text{not} P \text{ and } \text{not} Q
\]
(6) A propositional expression is equivalent to a disjunctive normal form or a conjunctive normal form:

For example, consider the propositional expression

\[ P_1 \text{ and not } (P_2 \text{ and } P_3) \]

It can be transformed into a disjunctive normal form by taking the following steps:

\[ P_1 \text{ and not } (P_2 \text{ and } P_3) \iff \\
( P_1 \text{ and not } P_2 ) \text{ or } (P_1 \text{ and not } P_3) \iff \\
P_1 \text{ and not } P_2 \text{ or } P_1 \text{ and not } P_3 \]
The boxed proof

Example: suppose we want to prove the validity (or truth) of the sequent:

\[ P \lor Q \vdash Q \lor P \]

Then what we need to do is to apply appropriate inference rules available in the propositional logic to deduce the conclusion

\[ Q \lor P \]

from the hypothesis

\[ P \lor Q \]
The boxed proof of the sequent

<table>
<thead>
<tr>
<th></th>
<th>from P or Q</th>
<th>h</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>from P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>infer P or Q [or-intro1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>infer Q or P [or-comm]</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>from Q</td>
<td></td>
</tr>
<tr>
<td></td>
<td>infer P or Q [or-intro2]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>infer Q or P [or-comm]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>infer Q or P [or-elim](h, 1, 2)</td>
<td></td>
</tr>
</tbody>
</table>
Exercise 2

1. Explain the notions:
   a. proposition
   b. conjunction
   c. disjunction
   d. negation
   e. implication
   f. equivalence
   g. tautology
   h. contradiction
   i. contingency
   j. sequent
   k. rule
   l. proof
2. Give a truth-table proof for the validity of each of the sequents:
   a. \( P, Q \vdash P \text{ and } Q \)
   b. \( P \text{ and } Q \vdash Q \)
   c. \( P \vdash P \text{ or } Q \)
   d. \( P \text{ or } Q, P \Rightarrow R, Q \Rightarrow R \vdash R \)
   e. \( P \vdash \neg \neg P \)
   f. \( Q \vdash P \Rightarrow Q \)
   g. \( P \Rightarrow Q, Q \Rightarrow P \vdash P \Leftrightarrow Q \)

3. Give a boxed proof for the truth of each of the sequents:
   a. \( P \text{ and } (Q \text{ and } R) \vdash (P \text{ and } Q) \text{ and } R \)
   b. \( P, Q, Q \Rightarrow R \vdash R \Rightarrow P \text{ and } R \)
   c. \( \neg (P \text{ or } Q) \vdash \neg Q \)
   d. \( P \text{ or } Q \vdash \neg (\neg P \text{ and } \neg Q) \)
4. Transform each of the following propositional expressions into a disjunctive normal form:

a. $P$ and not (not $Q$ and $R$)
b. $(P$ or $Q)$ and $(R$ or $W)$
c. not ($P$ $\Rightarrow$ $Q$) or (not $P$ and $Q$)
d. $P$ $\iff$ $Q$ and $Q$ $\iff$ $R$