Declarative Programming: Lecture Notes and Exercises

Part 1

Propositional formulas are built from elementary propositions, or “atoms,” and the logical constants \(\bot\) (false) and \(\top\) (true) using the connectives \(\neg\) (negation), \(\land\) (conjunction), \(\lor\) (disjunction), and \(\implies\) (implication). In this course, implications \(F \implies G\) will be usually written “backwards”: \(G \iff F\).

If one of the truth values \(false\), \(true\) is assigned to each atom then the truth value of a formula is defined as follows. The truth value of \(\bot\) is \(false\); the truth value of \(\top\) is \(true\). For propositional connectives, we use the truth tables:

<table>
<thead>
<tr>
<th></th>
<th>( F )</th>
<th>( \neg F )</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>( G )</th>
<th>( F \land G )</th>
<th>( F \lor G )</th>
<th>( F \implies G ) (or ( G \iff F ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>

Any set \(S\) of atoms can be thought of as an assignment of truth values to atoms: the atoms that belong to \(S\) get the value \(true\), and all other atoms get the value \(false\). If a formula \(F\) gets the value \(true\) for a truth assignment \(S\) then we way that \(S\) satisfies \(F\), or is a model of \(F\). For instance, the empty set \(\emptyset\) satisfies the formula

\[
p \iff q \land r
\]

because this formula gets the value \(true\) when all atoms \(p, q, r\) get the value \(false\).

**1.1.** Find all sets formed from the atoms \(p, q, r\) that do not satisfy formula (1).

**1.2.** Find a formula that is satisfied by \(\{p\}\) and by \(\{q\}\) but is not satisfied by \(\{p, q\}\).

We say that a set \(S\) of atoms is a model of a set \(\Gamma\) of formulas if \(S\) satisfies all formulas in \(\Gamma\).
1.3. Find all models of the set

\{p \leftarrow q \land r, q \leftarrow p, r \leftarrow p\}.

A tautology is a formula that is satisfied by every set of atoms.

1.4. Determine which of these formulas are tautologies.

(a) \(p \lor \top \leftarrow \neg q\).

(b) \(p \lor q \leftarrow \neg q \land \bot\).

(c) \(p \leftarrow q \land r \land \neg q\).

Two formulas or sets of formulas are equivalent to each other if they have the same models. For instance, the set \{p, q \leftarrow\} is equivalent to the formula \(p \land q\).

1.5. Determine whether the given formulas or sets of formulas are equivalent.

(a) \{p \leftarrow q, q \leftarrow r\} and \(p \leftarrow r\).

(b) \(p \leftarrow p\) and \(q \lor \neg q\).

(c) \{p \leftarrow p, q\} and \(q\).

(d) \{p, \neg p\} and \(\bot\).

(e) \{p, \neg p\} and \{p, \neg p, q \leftarrow r\}.

(f) \{p, \neg p \leftarrow q\} and \{p, \neg q\}.

(g) \neg p \leftarrow q and \neg q \leftarrow p.

(h) \neg p \leftarrow q and \neg (p \land q).

(i) \(p \leftarrow \neg q\) and \(q \leftarrow \neg p\).

(j) \(p \leftarrow \neg q\) and \(p \lor q\).

(k) \(\bot \leftarrow p\) and \(\neg p\).

(l) \{p \lor q, p \leftarrow q\} and \(p\).

(m) \{p \lor q, p \leftarrow q\} and \{p, q\}.