Predicate Logic
(SE 212 Tutorial 5)

Amin Bandali

Wed Oct 9, 2019
Today’s plan

- do some semantics questions from homework 4
- do some ND questions from homework 5
Today’s plan

• do some semantics questions from homework 4
• do some ND questions from homework 5
Provide a counterexample to show that the following argument is not valid and demonstrate that your answer is correct.

\[ \forall y : M \ . \ \exists x : N \ . \ p(g(x), y) \]

\[ \not\models \]

\[ \exists z : M \ . \ p(z, z) \]
Domain:
M = \{m_1, m_2\}
N = \{n_1, n_2\}

Mapping:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>g(.)</td>
<td>[ G(n_1) := m_1 ]</td>
</tr>
<tr>
<td></td>
<td>[ G(n_2) := m_2 ]</td>
</tr>
<tr>
<td>p(., .)</td>
<td>[ P(m_1, m_1) := F ]</td>
</tr>
<tr>
<td></td>
<td>[ P(m_1, m_2) := T ]</td>
</tr>
<tr>
<td></td>
<td>[ P(m_2, m_1) := T ]</td>
</tr>
<tr>
<td></td>
<td>[ P(m_2, m_2) := F ]</td>
</tr>
</tbody>
</table>
Premise: 

\[
\begin{align*}
&[\forall y : M . \exists x : N . p(g(x), y)] \\
&= [\exists x: N. p(g(x), ^\neg m1)] \text{ AND} \\
&[\exists x: N . p(g(x), ^\neg m2)] \\
&= (P(G(n1), m1) \text{ OR } P(G(n2), m1)) \text{ AND} \\
&(P(G(n1), m2) \text{ OR } P(G(n2), m2)) \\
&= (P(m1, m1) \text{ OR } P(m2, m1)) \text{ AND} \\
&(P(m1, m2) \text{ OR } P(m2, m2)) \\
&= (F \text{ OR } T) \text{ AND } (T \text{ OR } F) \\
&= T
\end{align*}
\]
Conclusion:

\[ \exists z : M \ . \ p(z, z) \]
\[ = P(m_1, m_1) \text{ OR } P(m_2, m_2) \]
\[ = F \text{ OR } F \]
\[ = F \]
Express the following sentences in predicate logic. Use types in your formalization. Is the set of formulas consistent? Demonstrate that your answer is correct using the semantics of predicate logic.

All programmer like some computers.
Some programmers use MAC.
Therefore, some people who like some computers use MAC.
All programmer like some computers.
Some programmers use MAC.
Therefore, some people who like some computers use MAC.

Formalization:

\[
\text{programmer}(x) \text{ means } x \text{ is a programmer} \\
\text{usesmac}(x) \text{ means } x \text{ uses MAC} \\
\text{likes}(x, y) \text{ means } x \text{ likes } y \\
\]

\[
\forall x: \text{Person} \ . \ \text{programmer}(x) \implies \\
\exists y: \text{Computer} \ . \ \text{likes}(x, y), \\
\exists x: \text{Person} \ . \ \text{programmer}(x) \land \text{usesmac}(x) \\
\]

\[
\vdash \\
\exists x: \text{Person} \ . \\
(\exists y: \text{Computer} \ . \ \text{likes}(x, y) \land \text{usesmac}(x))
\]
These sentences are consistent. Here is an interpretation in which all the formulas are T:

Domain:

- People = {John}
- Computer = {MacPro}

Mapping:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>programmer(.)</td>
<td>programmer(John) = T</td>
</tr>
<tr>
<td>likes(.,.)</td>
<td>likes(John, MacPro) = T</td>
</tr>
<tr>
<td>usesmac(.)</td>
<td>usesmac(John) = T</td>
</tr>
</tbody>
</table>
formula 1:
\[
\forall x: \text{Person} . \, \text{programmer}(x) \Rightarrow \exists y: \text{Computer} . \, \text{likes}(x, y)
\]
\[
= \left[ \text{programmer}(\neg \text{John}) \Rightarrow \exists y: \text{Computer} . \, \text{likes}(\neg \text{John}, y) \right]
\]
\[
= \text{programmer}(\text{John}) \, \text{IMP} \, \text{likes}(\text{John}, \text{MacPro})
\]
\[
= T \, \text{IMP} \, T
\]
\[
= T
\]

formula 2:
\[
\exists x: \text{Person} . \, \text{programmer}(x) \, \& \, \text{usesmac}(x)
\]
\[
= \text{programmer}(\text{John}) \, \text{AND} \, \text{usesmac}(\text{John})
\]
\[
= T \, \text{AND} \, T
\]
\[
= T
formula 3:

\[ [\exists x : \text{Person} . (\exists y : \text{Computer} . \\text{likes}(x, y) \& \text{usesmac}(x))] \]
\[ = [\exists y : \text{Computer} . \\text{likes}(\neg \text{John}, y) \& \text{usesmac}(\neg \text{John})] \]
\[ = \text{likes}(\text{John}, \text{MacPro}) \& \text{usesmac}(\text{John}) \]
\[ = T \& T \]
\[ = T \]
If the following arguments are valid, use natural deduction AND semantic tableaux to prove them; otherwise, provide a counterexample.

forall x . s(x) | t(x),
forall x . s(x) => t(x) & k(c, x),
forall x . t(x) => m(x)
|-
m(c)

where c is a constant
#check ND
forall x . s(x) | t(x),
forall x . s(x) => t(x) & k(c, x),
forall x . t(x) => m(x)
|-
m(c)
h05q01a (cont’d)

1) \(\forall x . s(x) \lor t(x)\) premise
2) \(\forall x . s(x) \Rightarrow t(x) \land k(c, x)\) premise
3) \(\forall x . t(x) \Rightarrow m(x)\) premise
4) \(s(c) \lor t(c)\) by \(\forall\)-elim on 1
5) \(s(c) \Rightarrow t(c) \land k(c, c)\) by \(\forall\)-elim on 2
6) \(t(c) \Rightarrow m(c)\) by \(\forall\)-elim on 3
7) case \(s(c)\) {
   8) \(t(c) \land k(c, c)\) by \(\text{imp}\)-elim on 5, 7
   9) \(t(c)\) by \(\text{and}\)-elim on 8
   10) \(m(c)\) by \(\text{imp}\)-elim on 6, 9
}
11) case \(t(c)\) {
    12) \(m(c)\) by \(\text{imp}\)-elim on 6, 11
}
13) \(m(c)\) by cases on 4, 7-10, 11-12
Is this formula a tautology?

\( \vdash (\exists x . \ p(x)) \implies (\forall y . \ p(y)) \)
No, this formula is not a tautology. Interpretation:

1) Domain = \{a, b\}

2) Mapping:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>p(.)</td>
<td>P(a) = T</td>
</tr>
<tr>
<td></td>
<td>P(b) = F</td>
</tr>
</tbody>
</table>

Conclusion:

\[ [(\exists x. \ p(x)) \rightarrow (\forall y. \ p(y))] \]
\[ = (P(a) \ OR \ P(b)) \ IMP \ (P(a) \ AND \ P(b)) \]
\[ = (T \ OR \ F) \ IMP \ (T \ AND \ F) \]
\[ = T \ IMP \ F \]
\[ = F \]
Is this argument valid?

\[
\forall x . \ p(x) \lor q(x), \quad \forall x . \ \neg p(x) \\
\vdash \quad \forall x . \ q(x)
\]
forall x . p(x) | q(x), forall x . !p(x) |- forall x . q(x)

1) forall x . p(x) | q(x) premise
2) forall x . !p(x) premise
3) for every xg {
   4) p(xg) | q(xg) by forall_e on 1
   5) case p(xg) {
       6) !p(xg) by forall_e on 2
       7) q(xg) by not_e on 5, 6
   }
   8) case q(xg) {} 
   9) q(xg) by cases on 4, 5-7, 8-8
}
10) forall x. q(x) by forall_i on 3-9
Announcements

- no tutorial next week (Oct 16) (reading week)
- no tutorial the week after (Oct 23) (midterm marking)