IN3070
IN4070

Autumn 2020

Exercises for the Course

Logic

Week 4



Exercise 3.6 (Semantics practice)

Give semantic proofs of the equivalences and implications on slide 33/41 in Lecture 4.

(If you want to be thorough, you should also prove that the statements "other direction is not valid" are true by giving a countermodel for the other direction).

Solution We give a solution to the final claim, as an example of semantic reasoning.

We need to show

$$\vDash \exists x(A(x) \wedge B(x)) \rightarrow (\exists x A(x) \wedge \exists x B(x))$$

Let an interpretation $\mathcal{I} = (D, \iota)$ be given, and assume that $\mathcal{I} \vDash \exists x(A(x) \wedge B(x))$ (or we are done).

Then there must exist a variable assignment α such that $v_{\mathcal{I}}(\alpha, (A(x) \wedge B(x))) = T$, i.e. such that $v_{\mathcal{I}}(\alpha, A(x))) = T$ and $v_{\mathcal{I}}(\alpha, B(x))) = T$.

But then $v_{\mathcal{I}}(\exists x A(x)) = T$ and $v_{\mathcal{I}}(\exists x B(x)) = T$, whence $\mathcal{I} \vDash \exists x A(x) \wedge \exists x B(x)$.

To show the converse does not hold, we need to show

$$\nvDash (\exists x A(x) \wedge \exists x B(x)) \rightarrow \exists x(A(x) \wedge B(x))$$

I.e. we have to find an $\mathcal{I}$ such that $\mathcal{I} \vDash \exists x A(x) \wedge \exists x B(x)$ but $\mathcal{I} \nvDash \exists x(A(x) \wedge B(x))$. But that is just a question of constructing an interpretation where the interpretations of A and B are non-empty disjoint sets. E.g. let the domain be $\mathbb{N}$ and A be (of the form $p(x)$ for p a unary predicate symbol) interpreted as the even numbers and B as the odd numbers.

Exercise 4.1 (Counter-models)

a) Let p be a 1-ary predicate symbol and s a 1-ary function symbol. Construct a falsifying interpretation for the sequent

$$\forall x\,(p(x) \vee p(s(x))),\ \forall x\,(\neg p(x) \vee \neg p(s(x))) \Longrightarrow$$

by beginning an LK derivation, and constructing an interpretation from an open branch.

Hint: start by using $\forall$-left on one of the two formulae with some dummy constant o, then continue using first one then the other of them, with larger and larger terms, until you see the pattern.

b) Try to find a falsifying interpretation $\mathcal{I} = (D, \iota)$ where the domain has only two elements, $|D| = 2$. This can *not* be read off from an open branch.

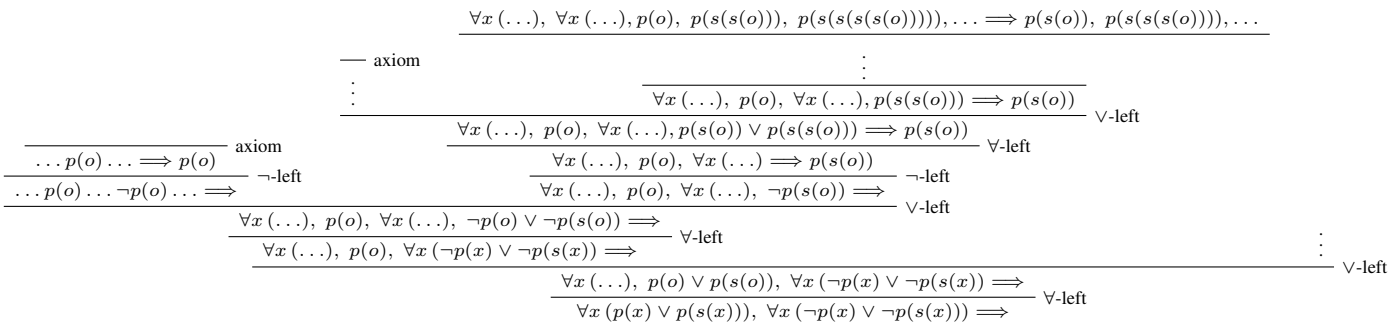
c) Is there a falsifying interpretation with only one domain element?

Solutions

a) Following is the beginning of such a proof tree. The open branch contains sequents

$$\forall x\,(\ldots),\ p(o),\ \forall x\,(\ldots), p(s(s(o))),\ p(s(s(s(s(o))))), \ldots \Longrightarrow p(s(o)),\ p(s(s(s(o)))), \ldots$$

such that the number of s's is even in the antecedent and odd in the succedent.



We can create an interpretation for this in the same way that we do for the completeness proof:

$$\begin{aligned} \mathcal{I} &= (D, \iota) \\ D &= \{\, o,\ s(o),\ s(s(o)),\ \ldots \} \\ s^\iota(x) &= s(x) \end{aligned}$$

Let p^ι be the smallest set such that

$$\begin{aligned} &o \in p^\iota, \text{and} \\ &\text{if } x \in p^\iota \text{ then } s(s(x)) \in p^\iota \end{aligned}$$

An interpretation with $p^\iota = \{s(o), s(s(s(o))), s(s(s(s(s(o))))), \ldots\}$, which can be found on another open branch, is just as correct.

b) Following is a falsifying interpretation with $|D| = 2$:

$$\begin{aligned} &\mathcal{I} = (D, \iota) && D = \{0, 1\} \\ &s^\iota = \{(0,1), (1,0)\} && p^\iota = \{0\} \end{aligned}$$

$p^\iota = \{1\}$ would also be correct. Up to renaming, any falsifying interpretation with $|D| = 2$ will be equivalent to this one.

c) No. Suppose $\mathcal{I}$ is a falsifying interpretation with one domain element a. Then $s^\iota = \{(a, a)\}$, and either

(i) $p^\iota = \emptyset$

(ii) $p^\iota = \{a\}$

If (i) holds, then $\mathcal{I} \not\models \forall x\,(p(x) \vee p(s(x)))$, since $a = s^\iota(a) \notin p^\iota$. If (ii) holds, then $\mathcal{I} \not\models \forall x\,(\neg p(x) \vee \neg p(s(x)))$, since $a = s^\iota(a) \in p^\iota$

So $\mathcal{I}$ is not a falsifying interpretation.

Exercise 4.2 **(Bernays-Schönfinkel)**

The Bernays-Schönfinkel class of formulas, named after Paul Bernays and Moses Schönfinkel, is a fragment of first-order logic formulas where satisfiability is decidable.

It is the class of all formulae of the shape

$$B = \exists x_1 \cdots \exists x_m \forall y_1 \cdots \forall y_n\, A$$

where A

- contains no quantifiers
- contains no function symbols (i.e. all terms are variables or constants)

Give an algorithm that decides the satsfiability of this kind of formula.

Hints:

- *a formula B is satisfiable if its negation $\neg B$ is not valid.*
- *consider the first few steps an LK proof for $\neg B$ must have.*
- *remember that there are no function symbols. How does the Herbrand universe of an open branch look?*
- *do we need to have infinite branches to achieve fairness?*

Solution

Using the first hint, we begin an LK proof for $\neg B$:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\sigma(\forall y_1 \cdots \forall y_n\, A) \Longrightarrow}\,\exists\text{-left}}{\vdots}}{\exists x_2 \cdots \exists x_m \forall y_1 \cdots \forall y_n\, A[x_1 \backslash o_1] \Longrightarrow}\,\exists\text{-left}}{\exists x_1 \cdots \exists x_m \forall y_1 \cdots \forall y_n\, A \Longrightarrow}\,\exists\text{-left}$$
$$\dfrac{\exists x_1 \cdots \exists x_m \forall y_1 \cdots \forall y_n\, A \Longrightarrow}{\Longrightarrow \neg \exists x_1 \cdots \exists x_m \forall y_1 \cdots \forall y_n\, A}\,\neg\text{-right}$$

with $\sigma = \{x_1 \backslash o_1, \ldots, x_m \backslash o_m\}$. Let the constants in A be $\{a_1, \ldots, a_k\}$. Since there are no function symbols, the Herbrand universe of the branch is currently the finite set $\mathcal{H} = \{o_1, \ldots, o_m, a_1, \ldots, a_k\}$.

The next step is to eliminate the universal quantifiers. Again we use ideas from the completeness proof to achieve this, substituting $y_1, \ldots, y_n$ by every constant in the Herbrand universe in a fair derivation. We need to try a maximum of $(m+k)^n$ substitutions on each branch, which is finite because m, k, and n are finite. If all branches are closed, then the formula is unsatisfiable. If the derivation is fair and the branch is not closed, then the formula is satisfiable.